

Oracle® Application Express

App Builder User's Guide



Release 18.2

F10919-01

November 2018

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Application Express App Builder User's Guide, Release 18.2

F10919-01

Copyright © 2003, 2018, Oracle and/or its affiliates. All rights reserved.

Primary Author: Terri Jennings

Contributors: Christina Cho, Hilary Farrell, Michael Hichwa, Joel Kallman, Sharon Kennedy, Harish Konakondla, David Peake, Anthony Rayner, Marc Sewtz, John Synders, Jason Straub, Patrick Wolf

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

| | |
|-----------------------------|-----|
| Audience | xl |
| Documentation Accessibility | xl |
| Related Documents | xl |
| Conventions | xli |

Changes in This Release

| | |
|--|-------|
| Changes in Oracle Application Express Release 18.2 | xlii |
| Changes in Oracle Application Express Release 18.1 | xliii |

1 Quick Start

| | |
|---|------|
| About Accessing Your Development Environment | 1-2 |
| Understanding Oracle Application Express | 1-2 |
| What is Oracle Application Express? | 1-3 |
| How Oracle Application Express Works | 1-3 |
| About the Differences Between Runtime and Full Development Environments | 1-3 |
| Understanding the Workspace Home Page | 1-4 |
| Workspace Home Page | 1-4 |
| Top Apps and Top Users Dashboards | 1-6 |
| News and Messages Region | 1-6 |
| Developer Navigation Tools | 1-7 |
| Global Search | 1-7 |
| Administration Menu | 1-10 |
| Help Menu | 1-11 |
| Account Menu | 1-12 |
| Available Updates Region | 1-13 |
| Understanding App Builder | 1-13 |
| Accessing App Builder | 1-14 |
| Understanding the App Builder Home Page | 1-15 |
| App Builder Home Page Icons and Navigation Bar | 1-16 |
| Reset and Create Buttons | 1-18 |

| | |
|--|------|
| Recent List, Tasks List, and Migrations Link | 1-19 |
| Workspace Utilities | 1-20 |
| Viewing Cross Application Reports | 1-21 |
| Accessing the Application Home Page | 1-22 |
| Understanding the Application Home Page | 1-22 |
| Edit Application Properties Button | 1-23 |
| Application Home Page Icons | 1-24 |
| Create Page Button | 1-25 |
| Application Home Page Navigation Bar | 1-25 |
| Application Home Page Tasks List | 1-27 |
| Recently Edited Pages | 1-27 |
| Changing Your Profile or Password | 1-27 |
| Editing User Preferences | 1-28 |
| Using Oracle Application Express Documentation | 1-29 |
| Accessing Oracle Help Center | 1-29 |
| Searching the Oracle Help Center | 1-30 |
| Downloading the Documentation Library | 1-31 |
| Viewing Field-Level Help | 1-31 |
| Viewing Field-level Help in Oracle Application Express | 1-31 |
| Viewing Help in Page Designer | 1-32 |

2 App Builder Concepts

| | |
|---|-----|
| Understanding Oracle Application Express Applications | 2-2 |
| What is an Oracle Application Express Application? | 2-2 |
| What is the Difference Between a Database and Websheet Application? | 2-2 |
| Running an Application or Page | 2-3 |
| Run Application and Run Page Buttons | 2-4 |
| How Your Browser Impacts the Way Applications Run | 2-4 |
| Running an Application from the App Builder Home Page | 2-5 |
| Running an Application from the Application Home Page | 2-5 |
| Running a Page on the Application Home Page | 2-6 |
| How Does Page Processing and Page Rendering Work? | 2-6 |
| How the Application Express Engine Renders and Processes Pages | 2-6 |
| About Implicit Commit Points | 2-7 |
| Understanding Conditional Rendering and Processing | 2-7 |
| What is a Condition? | 2-7 |
| Selecting a Condition | 2-8 |
| Condition Example: Current Page in Expression 1 | 2-8 |
| Condition Example: Exists | 2-8 |
| Condition Example: PL/SQL Expression | 2-8 |

| | |
|---|------|
| About Verifying User Identity | 2-9 |
| About Controlling Access to Controls and Components | 2-9 |
| About Enabling Support for Bookmarks | 2-9 |
| Understanding Session State Management | 2-10 |
| What is Session State? | 2-10 |
| About Session IDs | 2-10 |
| What Is a Session? | 2-11 |
| Viewing Session State | 2-11 |
| Understanding URL Syntax | 2-13 |
| About the Application Page URL | 2-13 |
| About Using f?p Syntax to Link Pages | 2-14 |
| Calling a Page Using an Application Alias or Page Alias | 2-16 |
| Calling a Page from a Button URL | 2-16 |
| Managing Session State Values | 2-17 |
| About Referencing Session State | 2-17 |
| About Setting Session State | 2-18 |
| Clearing Session State | 2-19 |
| About Clearing Cache for an Item | 2-19 |
| About Clearing Cache for All Page Items | 2-19 |
| Clearing Cache for an Entire Application | 2-20 |
| About Resetting an Application Completely | 2-20 |
| About Clearing Cache for the Current User Session | 2-21 |
| Referencing Session State Using Bind Variable Syntax | 2-21 |
| About Using Bind Variable Syntax | 2-21 |
| About Using Bind Variables in Regions Based on a SQL Query or LOV | 2-21 |
| About Using Bind Variables in Regions Based on PL/SQL | 2-22 |
| About Session Cloning | 2-22 |
| Using Substitution Strings | 2-23 |
| About Using Substitution Strings | 2-23 |
| Controlling Output Escaping in Substitution Strings | 2-25 |
| About Determining Substitution String Usage within Templates | 2-26 |
| Using Built-in Substitution Strings | 2-26 |
| APEX\$ROW_NUM | 2-28 |
| APEX\$ROW_SELECTOR | 2-28 |
| APEX\$ROW_STATUS | 2-28 |
| APP_ID | 2-29 |
| APP_ALIAS | 2-29 |
| APP_AJAX_X01, ... APP_AJAX_X10 | 2-30 |
| APP_BUILDER_SESSION | 2-30 |
| APP_DATE_TIME_FORMAT | 2-31 |
| APP_IMAGES | 2-31 |

| | |
|--|------|
| APP-NLS_DATE_FORMAT | 2-31 |
| APP-NLS_TIMESTAMP_FORMAT | 2-32 |
| APP-NLS_TIMESTAMP_TZ_FORMAT | 2-32 |
| APP_PAGE_ALIAS | 2-33 |
| APP_PAGE_ID | 2-33 |
| APP_REQUEST_DATA_HASH | 2-33 |
| APP_SESSION | 2-34 |
| APP_SESSION_VISIBLE | 2-34 |
| APP_TITLE | 2-35 |
| APP_UNIQUE_PAGE_ID | 2-35 |
| APP_USER | 2-36 |
| AUTHENTICATED_URL_PREFIX | 2-37 |
| BROWSER_LANGUAGE | 2-37 |
| CURRENT_PARENT_TAB_TEXT | 2-37 |
| DEBUG | 2-38 |
| HOME_LINK | 2-38 |
| IMAGE_PREFIX | 2-39 |
| JET_BASE_DIRECTORY | 2-39 |
| JET_CSS_DIRECTORY | 2-39 |
| JET_JS_DIRECTORY | 2-40 |
| LOGIN_URL | 2-40 |
| LOGOUT_URL | 2-40 |
| APP_TEXT\$Message_Name, APP_TEXT\$Message_Name\$Lang | 2-41 |
| PRINTER_FRIENDLY | 2-41 |
| PROXY_SERVER | 2-41 |
| PUBLIC_URL_PREFIX | 2-42 |
| REQUEST | 2-42 |
| Using REQUEST | 2-42 |
| SCHEMA_OWNER | 2-44 |
| SQLERRM | 2-44 |
| SYSDATE_YYYYMMDD | 2-45 |
| THEME_DB_IMAGES | 2-45 |
| THEME_IMAGES | 2-45 |
| WORKSPACE_IMAGES | 2-45 |
| WORKSPACE_ID | 2-46 |

3 Utilizing the App Gallery

| | |
|---|-----|
| About App Gallery | 3-1 |
| Managing Productivity and Sample Apps in a Full Development Environment | 3-2 |
| Accessing the App Gallery Page | 3-3 |

| | |
|--|------|
| Installing a Productivity and Sample App | 3-4 |
| Running a Productivity and Sample App | 3-5 |
| Removing a Productivity and Sample App | 3-5 |
| Changing Authentication for an Installed Productivity and Sample App | 3-6 |
| Unlocking an Installed Productivity Application | 3-6 |
| Updating a Productivity and Sample App | 3-7 |
| Understanding Sample Database Application | 3-8 |
| Installing and Running Sample Database Application | 3-8 |
| Reviewing Sample Database Application | 3-9 |
| Home Page | 3-10 |
| About the Navigation Menu | 3-11 |
| Customers Page | 3-12 |
| Using the Customers Page | 3-12 |
| Products Page | 3-14 |
| Using the Products Page | 3-15 |
| Orders Page | 3-15 |
| Using the Orders Page | 3-16 |
| Reports Page | 3-17 |
| Administration Page | 3-18 |
| Modifying Sample Database Application | 3-19 |
| Why Modify a Sample Database Application? | 3-19 |
| Editing Sample Database Application | 3-20 |
| Understanding AnyCo IT Department Websheet | 3-22 |
| Installing and Running AnyCo IT Department | 3-22 |
| Understanding Websheet Navigation | 3-23 |
| Understanding AnyCo IT Department | 3-25 |
| Navigating Between Pages and Viewing Page Sections | 3-25 |
| Home Page | 3-26 |
| Projects Page | 3-29 |
| Project Review Page | 3-30 |
| Systems Page | 3-30 |
| Planned Purchase Review | 3-31 |
| Systems Budget Review | 3-33 |
| Modifying Websheet Content | 3-35 |
| Why Modify Sample Websheet Application? | 3-35 |
| Updating a Websheet Section | 3-35 |
| Utilizing App Administration | 3-37 |
| Making a Service Request | 3-38 |
| Setting Workspace Preferences | 3-39 |
| Managing Users | 3-39 |
| Managing Saved Interactive Reports | 3-40 |

| | |
|---|------|
| Managing Interactive Report Subscriptions | 3-41 |
| Viewing Activity Reports | 3-41 |
| Viewing Page Views | 3-41 |
| Viewing Top Users | 3-41 |
| Using a Productivity and Sample Apps Only Workspace | 3-42 |
| What Is a Productivity and Sample Apps Only Workspace? | 3-42 |
| Requesting a Productivity and Sample Apps Only Workspace | 3-42 |
| Signing In To a Productivity and Sample Apps Only Workspace | 3-43 |
| Using the App Page | 3-43 |

4 Managing the Application Life Cycle with Team Development

| | |
|---|------|
| Accessing Team Development | 4-2 |
| Team Development Home Page | 4-2 |
| Tracking Features | 4-4 |
| Viewing the Features Report | 4-4 |
| Creating a Feature | 4-5 |
| About Feature Email Notifications | 4-6 |
| Updating or Deleting a Feature | 4-6 |
| About Creating Parent Features | 4-7 |
| How to Attach Files to a Feature | 4-7 |
| Attaching a File to a Feature | 4-8 |
| Viewing, Editing, and Removing Attached Files | 4-8 |
| Viewing Other Feature Reports | 4-8 |
| Viewing the Features Tree | 4-9 |
| Viewing the Features Calendar | 4-10 |
| Viewing The Feature History Report | 4-10 |
| Viewing the Feature Progress Log | 4-11 |
| Viewing the Feature Focus Areas Chart | 4-11 |
| Viewing the Approval Statuses | 4-12 |
| Viewing the Feature Owners Chart | 4-12 |
| Assigning or Updating Milestones Associated with Features | 4-12 |
| Tracking Milestones | 4-13 |
| Viewing the Milestones Report | 4-13 |
| Creating a Milestone | 4-13 |
| Updating or Deleting a Milestone | 4-14 |
| Viewing Other Milestone Reports | 4-14 |
| Viewing the Milestone Calendar | 4-14 |
| Viewing the Milestones By Owner Report | 4-15 |
| Viewing the Features by Milestone Report | 4-15 |
| Tracking To Dos | 4-15 |

| | |
|---|------|
| Viewing To Dos | 4-16 |
| Creating a To Do | 4-16 |
| Updating a To Do | 4-17 |
| Managing Files Attached to a To Do | 4-17 |
| Attaching a File to a To Do | 4-17 |
| Viewing, Editing, and Removing Attached Files | 4-18 |
| Viewing Other To Do Reports | 4-18 |
| Viewing the To Dos Calendar | 4-18 |
| Viewing the To Do Progress Log | 4-19 |
| Managing Bugs | 4-19 |
| Viewing Bugs | 4-20 |
| Creating a Bug | 4-20 |
| Editing a Bug | 4-20 |
| Managing Files Attached to a Bug | 4-21 |
| Attaching a File to a Bug | 4-21 |
| Viewing, Editing, and Removing Attached Files | 4-22 |
| Viewing Other Bug Reports | 4-22 |
| Viewing the Bug Calendar | 4-22 |
| Viewing the Bugs By Developer | 4-23 |
| Viewing the Bugs By Day | 4-23 |
| Managing Feedback | 4-24 |
| About Feedback | 4-24 |
| Adding Feedback Using the Create Application Wizard | 4-25 |
| Adding Feedback Using the Create Page Wizard | 4-26 |
| Submitting Feedback Within a Running Application | 4-27 |
| Reviewing Feedback Within an Application | 4-27 |
| Configuring Feedback Within an Application | 4-28 |
| Tracking Feedback Team in Development | 4-28 |
| Accessing the Feedback Report in Team Development | 4-28 |
| Viewing Feedback by Filing User | 4-29 |
| Viewing the Feedback Calendar | 4-29 |
| Managing Team Development Utilities | 4-30 |
| Accessing Team Development Utilities | 4-31 |
| Configuring Team Development Settings | 4-32 |
| Viewing a Release Summary | 4-33 |
| Managing Feature Utilities | 4-33 |
| Renaming Focus Areas | 4-34 |
| Downloading File Attachments | 4-34 |
| Purging Data | 4-34 |
| Managing News Entries | 4-35 |

5 Creating Websheet Applications

| | |
|---|------|
| Creating a Websheet Application | 5-2 |
| Creating a Websheet from App Builder | 5-2 |
| Creating a Websheet Application from a Running Websheet | 5-3 |
| Copying an Existing Websheet Page | 5-4 |
| Running a Websheet | 5-4 |
| Running a Websheet from the App Builder Home Page | 5-5 |
| Running a Websheet from the Application Home Page | 5-5 |
| Determining a Websheet URL | 5-6 |
| Editing Page Details | 5-6 |
| Understanding Websheet Properties | 5-7 |
| Editing Websheet Properties | 5-7 |
| Websheet Properties | 5-7 |
| Websheet | 5-8 |
| Details | 5-8 |
| Logo | 5-9 |
| Style | 5-9 |
| Authentication | 5-9 |
| Authorization | 5-10 |
| SQL | 5-11 |
| Deleting a Websheet Application | 5-11 |
| Viewing the Websheet Dashboard | 5-11 |
| Monitoring Websheet Activity | 5-12 |
| Controlling Websheet Access | 5-13 |
| How Websheet Access Works | 5-13 |
| Specifying the Access Control List Type | 5-13 |
| Creating Access Control List Entries | 5-14 |

6 Creating Database Applications

| | |
|--|------|
| About the Create Application Wizard | 6-2 |
| Understanding Page Types, Features, and Settings | 6-4 |
| About Creating a New Database Application | 6-5 |
| Available Page Types in the Create Application Wizard | 6-5 |
| Available Features in the Create Application Wizard | 6-7 |
| Available Settings in the Create Application Wizard | 6-8 |
| About Removing Features Created with the Create Application Wizard | 6-11 |
| Creating a Database Application Based on a Table or Query | 6-12 |

| | |
|--|------|
| Creating a Database Application from a Spreadsheet | 6-13 |
| About Creating an Application from a Spreadsheet | 6-14 |
| Loading Spreadsheet Data from a File | 6-14 |
| Loading Spreadsheet Data by Copying and Pasting | 6-17 |
| Using Application Blueprints | 6-19 |
| What Is an Application Blueprint? | 6-19 |
| Viewing an Application Blueprint | 6-20 |
| Loading an Application Blueprint | 6-20 |
| Copying a Database Application | 6-21 |
| Copying a Database Application Using the Create Application Wizard | 6-21 |
| Copying an Application from the Application Home Page | 6-22 |
| Creating Applications for Mobile Devices | 6-22 |
| About Mobile Application Design | 6-22 |
| About Mobile Support | 6-23 |
| Creating Mobile Applications | 6-24 |
| Deleting an Application | 6-25 |
| Deleting an Application from the Application Home Page | 6-25 |
| Deleting an Application from the Edit Application Definition Page | 6-25 |
| Managing Application Attributes | 6-26 |
| Editing Application Attributes | 6-26 |
| Accessing the Edit Application Definition Page | 6-27 |
| Edit Application Definition Page | 6-27 |
| Configuring Security Attributes | 6-35 |
| Accessing the Edit Security Attributes Page | 6-36 |
| Security Attributes Page | 6-36 |
| Configuring Globalization Attributes | 6-44 |
| Accessing the Globalization Attributes Page | 6-44 |
| Edit Globalization Attributes Page | 6-45 |
| Managing the Application User Interface | 6-48 |
| Accessing User Interface Attributes | 6-49 |
| User Interface Page | 6-49 |
| Defining an Application Logo | 6-52 |
| Adding Desktop UI to a Mobile UI Only Application | 6-53 |
| Editing User Interface Details | 6-54 |
| User Interface Details Page | 6-55 |
| Adding Developer Comments | 6-59 |
| Developer Comment, Bug, or To Do Button | 6-60 |
| Adding Developer Comments to an Application | 6-60 |
| Viewing and Editing Developer Comments | 6-60 |
| Deleting Developer Comments | 6-61 |
| Deleting a Specific Developer Comment | 6-61 |

| | |
|---|------|
| Deleting Multiple Developer Comments | 6-61 |
| Accessing the Developer Comments from Shared Components | 6-62 |
| Viewing the Developer Comments Calendar | 6-62 |
| Using the Find Icon | 6-63 |
| Accessing the Find Icon | 6-63 |
| Searching for Items | 6-64 |
| Searching for Pages | 6-65 |
| Searching for Queries | 6-65 |
| Searching for Tables | 6-65 |
| Using the PL/SQL Finder | 6-66 |
| Searching for Images | 6-67 |
| Adding Database Application Comments | 6-67 |
| Managing Application Groups | 6-68 |
| Viewing Database Application Groups | 6-68 |
| Creating a Database Application Group | 6-68 |
| Assigning a Database Application to an Application Group | 6-69 |
| Removing a Database Application from an Application Group | 6-69 |
| Deleting an Database Application Group | 6-69 |
| Using Application Utilities | 6-70 |
| Application Utilities | 6-71 |
| Accessing the Utilities Page | 6-72 |
| Viewing the Application Dashboard | 6-73 |
| Upgrading an Application to Include New Components | 6-74 |
| About Upgrading to Include New Components | 6-74 |
| Running Upgrade Application | 6-74 |
| Viewing the Recently Updated Pages Report | 6-74 |
| Viewing the Change History Report | 6-75 |
| Using the Database Object Dependencies Report | 6-75 |
| Accessing Application Express Views | 6-76 |
| Accessing Page Specific Utilities | 6-76 |
| Accessing Cross Page Utilities | 6-77 |
| Using Interactive Report Utilities | 6-78 |
| Deleting Saved Interactive Reports | 6-78 |
| Viewing Subscriptions | 6-79 |

7 Managing Pages in a Database Application

| | |
|--|-----|
| About Creating New Pages | 7-1 |
| Adding a New Page to an Application | 7-2 |
| Understanding Page Types in the Create Page Wizard | 7-4 |
| Available Component Page Types | 7-4 |

| | |
|--|------|
| Available Features Page Types | 7-6 |
| Available Components for Mobile User Interface Applications | 7-8 |
| About Removing Features Created with the Create Page Wizard | 7-10 |
| Creating Dialog Pages | 7-10 |
| About Supported Dialogs | 7-10 |
| Using a Dialog Page Template | 7-11 |
| Creating a Dialog Page Template | 7-11 |
| About Defining Dialog Page Template Attributes | 7-12 |
| Defining a Dialog Page as a in Page Designer | 7-12 |
| Generating a Cancel Dialog Dynamic Action on the Dialog Page | 7-13 |
| Generating a Close Dialog Page Process on the Dialog Page | 7-13 |
| About Branching with Dialog Pages | 7-14 |
| Handling Refresh Actions on Parent Page | 7-14 |
| About Copying Database Application Pages | 7-15 |
| About Deleting Database Application Pages | 7-16 |
| Creating Page Groups | 7-16 |
| Viewing Page Groups | 7-17 |
| Creating a Page Group | 7-17 |
| Editing a Page Group Definition | 7-18 |
| Assigning a Page to a Page Group | 7-18 |
| Viewing Pages by Page Group | 7-18 |
| Deleting a Page Group | 7-19 |
| Locking and Unlocking a Page | 7-19 |
| Determining If a Page Is Locked | 7-19 |
| Locking a Page | 7-20 |
| Locking a Page from the Application Home Page | 7-20 |
| Locking a Page from the Page Locks Page | 7-20 |
| Locking a Page in Page Designer | 7-21 |
| Unlocking a Page | 7-21 |
| Unlocking Pages from Utilities | 7-21 |
| Unlocking Pages from the Application Home Page | 7-22 |
| Unlocking a Page in Page Designer | 7-22 |
| Unlocking Pages As a Workspace Administrator | 7-22 |

8 Editing Pages in Page Designer

| | |
|---|------|
| About Page Designer | 8-2 |
| Understanding Page Designer UI Elements | 8-3 |
| Page Designer Toolbar | 8-4 |
| Left Pane of Page Designer | 8-7 |
| Central Pane of Page Designer | 8-10 |

| | |
|---|------|
| Layout Tab | 8-11 |
| Gallery | 8-13 |
| Component View Tab | 8-14 |
| Messages Tab | 8-15 |
| Page Search Tab | 8-16 |
| Help Tab | 8-16 |
| Right Pane of Page Designer (Property Editor) | 8-17 |
| Code Editor | 8-18 |
| Using Page Designer | 8-20 |
| Viewing a Page in Page Designer | 8-20 |
| Searching for Pages | 8-21 |
| Dragging and Dropping Tabs Between Panes | 8-22 |
| Switching Between Three Pane and Two Pane Mode | 8-23 |
| Accessing Property Editor | 8-24 |
| Adding a Region, Item, or Button by Dragging and Dropping | 8-25 |
| About Adding Components Using Gallery Context Menus | 8-26 |
| Adding a Region, Item, or Button from the Rendering Tab | 8-27 |
| Adding a Region, Item, or Button from the Component View Tab | 8-28 |
| Copying a Region, Item, or Button | 8-29 |
| Creating Pre-Rendering Branches, Computations, and Processes | 8-30 |
| Creating Page Submission Branches, Validations, Computations, and Processes | 8-30 |
| Managing Page Attributes | 8-31 |
| Editing Page Attributes | 8-32 |
| Incorporating JavaScript Using Page Designer | 8-32 |
| Configuring Rejoin Sessions | 8-34 |
| Configuring Dialog Attributes | 8-35 |
| Configuring Page Attributes to Warn Users of Unsaved Changes | 8-36 |
| Running a Page from Page Designer | 8-37 |
| Runtime Developer Toolbar | 8-38 |
| Copying a Database Application Page | 8-39 |
| Deleting Pages | 8-40 |
| Deleting a Page | 8-40 |
| Deleting Multiple Pages | 8-41 |
| Deleting a Range of Pages | 8-41 |

9 Developing Reports

| | |
|----------------------------|-----|
| Understanding Report Types | 9-2 |
| Interactive Grid | 9-2 |
| Interactive Report | 9-4 |
| Classic Report | 9-5 |

| | |
|---|------|
| Report and Form | 9-6 |
| Creating a Report Using the Create Application Wizard | 9-6 |
| Creating an Interactive or Classic Report Using the Create Application Wizard | 9-7 |
| Creating an Interactive Grid Using the Create Application Wizard | 9-8 |
| Report Options When Running the Create Application Wizard | 9-10 |
| Creating a Report Using the Create Page Wizard | 9-11 |
| Creating a Report on a New Page | 9-11 |
| Creating a Classic or Interactive Report on New Page Using Remote Database References | 9-12 |
| Report Options When Running the Create Page Wizard | 9-14 |
| Creating a Report by Dragging and Dropping from the Gallery | 9-17 |
| Managing Interactive Grids | 9-18 |
| Managing Interactive Grid Region Attributes | 9-19 |
| Editing Interactive Grid Region Attributes | 9-20 |
| Creating Master Detail from an Existing Interactive Grid | 9-21 |
| Displaying Interactive Grids Conditionally | 9-24 |
| Managing Interactive Grid Attributes | 9-25 |
| Editing Interactive Grid Attributes | 9-26 |
| Making an Existing Interactive Grid Editable | 9-26 |
| Controlling Interactive Grid Pagination | 9-28 |
| Customizing the Interactive Grid Toolbar | 9-29 |
| Enabling Users to Save Public Reports in Interactive Grids | 9-30 |
| Restricting Who Can Save Public Reports | 9-31 |
| Enabling Icon View in an Interactive Grid | 9-32 |
| Enabling Detail View in an Interactive Grid | 9-35 |
| Managing Interactive Grid Column Attributes | 9-36 |
| Editing Common Grid Column Attributes | 9-37 |
| Defining a Column as a List of Values for Filtering | 9-38 |
| Creating a Column Link in an Interactive Grid | 9-39 |
| About Customizing Interactive Grids in a Running Application | 9-41 |
| Saving Interactive Grids | 9-41 |
| How the User Type Effects Available Save Options | 9-42 |
| About Configuration Dependencies When Saving Interactive Grids | 9-42 |
| Saving a Private Interactive Grid | 9-43 |
| Saving a Public Interactive Grid | 9-43 |
| Saving or Renaming a Primary Interactive Grid | 9-44 |
| Saving an Alternative Interactive Grid | 9-44 |
| Renaming a Saved Interactive Grid Report | 9-45 |
| Deleting a Saved Interactive Grid Report | 9-45 |
| Resetting a Saved Interactive Grid Report | 9-45 |
| Managing Interactive Reports | 9-46 |

| | |
|--|------|
| Managing Interactive Report Region Attributes | 9-47 |
| Editing Interactive Report Region Attributes | 9-47 |
| Displaying Interactive Reports Conditionally | 9-48 |
| Managing Interactive Report Attributes | 9-49 |
| Editing Interactive Report Attributes | 9-49 |
| Managing Link Columns | 9-50 |
| Customizing the Interactive Report Search Bar | 9-54 |
| Customizing the Actions Menu | 9-57 |
| Configuring Actions Menu Download Options | 9-58 |
| Controlling Interactive Report Pagination | 9-63 |
| Enabling Icon View | 9-63 |
| Enabling Detail View | 9-66 |
| Configuring Advanced Attributes for Interactive Reports | 9-67 |
| Managing Interactive Report Column Attributes | 9-69 |
| Editing Interactive Report Column Attributes | 9-70 |
| Creating a Column Link in an Interactive Report | 9-71 |
| Defining a Column as a List of Values in an Interactive Report | 9-72 |
| Formatting Columns in an Interactive Report | 9-73 |
| About Filtering on Conditional Links in Interactive Report Columns | 9-74 |
| Customizing Interactive Reports in a Running Application | 9-74 |
| About the Search Bar | 9-75 |
| About the Actions Menu | 9-76 |
| Using Interactive Report Filters | 9-76 |
| Saving Interactive Reports | 9-77 |
| Linking to Interactive Reports | 9-83 |
| Request Syntax | 9-84 |
| Name and Value Syntax | 9-84 |
| ClearCache Syntax | 9-85 |
| Restrictions When Linking to Interactive Reports | 9-86 |
| Valid Linking Examples | 9-86 |
| Invalid Linking Examples | 9-86 |
| Linking to Shared Interactive Reports | 9-87 |
| Managing Classic Reports | 9-89 |
| Managing Classic Report Region Attributes | 9-89 |
| Editing Classic Report Region Attributes | 9-89 |
| Managing Classic Report Attributes | 9-90 |
| Editing Classic Report Attributes | 9-91 |
| Editing Classic Report Pagination | 9-92 |
| Adding Support for XML File or a CSV File Export | 9-93 |
| Enabling Classic Report CSV Output Option | 9-93 |
| Controlling Classic Report Column Breaks | 9-94 |

| | |
|---|-------|
| Managing Classic Report Column Attributes | 9-95 |
| Altering Classic Report Column Layout | 9-96 |
| Enabling Column Sorting in a Classic Report | 9-97 |
| Creating a Column Link in a Classic Report | 9-98 |
| Defining an Updatable Column in a Classic Report | 9-99 |
| Defining a Column as a List of Values in a Classic Report | 9-100 |
| Controlling When Classic Report Columns Display | 9-101 |
| Adding a Download Link to a Classic Report | 9-102 |
| Printing Report Regions | 9-102 |
| About Printing Reports to PDF | 9-103 |
| About Report Printing Configuration Options | 9-103 |
| About Classic Report Printing Methods | 9-104 |
| Configuring Classic Report Region Print Attributes | 9-105 |
| Configuring a Classic Report Print Attributes | 9-105 |
| Configuring Interactive Report Region Print Attributes | 9-106 |
| Configuring Interactive Report Regions for Printing | 9-106 |
| About Report Queries | 9-107 |
| About Report Layouts | 9-107 |
| Understanding BLOB Support in Forms and Reports | 9-107 |
| About BLOB Support in Reports | 9-108 |
| About Providing a Download Link | 9-108 |
| About Displaying the BLOB | 9-109 |
| About Working With BLOBs Procedurally | 9-110 |

10 Developing Forms

| | |
|--|-------|
| Understanding Form Types | 10-1 |
| Editable Interactive Grid | 10-2 |
| Form on a Table | 10-2 |
| Master Detail Forms | 10-3 |
| Creating a Form | 10-4 |
| Creating a Form Using the Create Page Wizard | 10-4 |
| Form Options When Running the Create Page Wizard | 10-6 |
| Creating a Form Using the Create Application Wizard | 10-8 |
| Creating a Form Region | 10-10 |
| About Making an Interactive Grid Editable | 10-10 |
| Managing Master Detail Forms | 10-11 |
| About Master Detail Forms | 10-11 |
| Creating a Master Detail Form Using the Create Application Wizard | 10-15 |
| Creating a Side by Side Master Detail Using the Create Page Wizard | 10-17 |
| Creating a Stacked Master Detail Using the Create Page Wizard | 10-18 |

| | |
|---|-------|
| Creating a Two Page Drill Down Master Detail Using the Create Page Wizard | 10-20 |
| Validating User Input in Forms | 10-21 |
| About Creating Validations and Error Messages | 10-21 |
| Associating a Validation with a Specific Item | 10-22 |
| About the Error Message Display Location | 10-22 |
| Understanding BLOB Support in Forms | 10-22 |
| About BLOB Support in Forms | 10-23 |
| About Uploading and Downloading Files into a BLOB | 10-23 |
| About Displaying the BLOB | 10-24 |
| About Removing the Image Reference | 10-24 |

11 Managing Database Application Components

| | |
|--|-------|
| Understanding Oracle JET Integration with Oracle Application Express | 11-1 |
| About Oracle JET | 11-2 |
| How Oracle JET is integrated with Oracle Application Express | 11-2 |
| Switching from AnyChart to JET Chart | 11-4 |
| Upgrading to JET Chart Using Application Upgrade Utility | 11-4 |
| Searching a Workspace for AnyChart Using SQL Commands | 11-4 |
| Creating Calendars | 11-6 |
| About Supported Calendars | 11-6 |
| Sample Calendar | 11-8 |
| Adding a Calendar to a New Page Using a Local Database | 11-9 |
| Adding a Calendar on a New Page Using Remote Database References | 11-10 |
| Creating a Calendar in Page Designer | 11-12 |
| Managing Calendar Attributes | 11-14 |
| Editing Calendar Attributes in the Property Editor | 11-14 |
| Editing an Existing Calendar to Include Add and Edit Functionality | 11-15 |
| Adding Calendar Initialization JavaScript Code | 11-17 |
| Enabling the Dragging and Dropping of Data In an Existing Calendar | 11-18 |
| About Dynamic Action Support for Calendar | 11-19 |
| Creating Maps | 11-20 |
| About Map Chart Support | 11-21 |
| About Creating SQL Queries for Maps | 11-21 |
| Adding a Map Chart to a New Page | 11-24 |
| Adding a Map Chart in Page Designer | 11-25 |
| Managing Map Charts | 11-26 |
| Editing Map Chart Attributes | 11-26 |
| Using Custom XML with Map Charts | 11-27 |
| Enabling Automatic Updates | 11-28 |
| Creating Charts | 11-29 |

| | |
|---|-------|
| About Supported Charts | 11-29 |
| Sample Charts | 11-30 |
| Adding a Chart by Running the Create Page Wizard | 11-31 |
| Adding a Chart on a New Page Using Remote Database References | 11-32 |
| Adding a Chart in Page Designer | 11-35 |
| Managing Charts | 11-36 |
| Editing Chart Attributes | 11-37 |
| Switching Chart Type | 11-37 |
| Enabling Automatic Refresh | 11-38 |
| Using Custom JavaScript with Charts | 11-40 |
| Creating Help for Your Application | 11-41 |
| Creating Help | 11-41 |
| About Creating Help | 11-42 |
| Creating a Help Page and Region | 11-42 |
| Defining Page Help | 11-43 |
| Defining Item Help | 11-44 |
| Using the Bulk Edit Item Help Report | 11-44 |
| Seeding Item Help Topics | 11-45 |
| Creating a Help Navigation Bar Entry | 11-45 |

12 Using Themes and Theme Styles

| | |
|--|-------|
| Using Themes | 12-1 |
| About Themes | 12-2 |
| About Responsive Design and the Universal Theme | 12-2 |
| Using Theme Styles and Theme Roller | 12-4 |
| About Theme Styles | 12-4 |
| Creating a Theme Style | 12-4 |
| Editing a Theme Style from the Themes Page | 12-5 |
| Editing a Theme Style from the User Interface Page | 12-5 |
| Enabling Users to Select a Theme Style | 12-6 |
| Using Theme Roller | 12-6 |
| Understanding Template Options | 12-9 |
| About Template Options | 12-9 |
| About Default Template Options | 12-10 |
| About Template Option Groups and Presets | 12-10 |
| Accessing the Themes Page | 12-10 |
| Editing Themes | 12-11 |
| Editing a Theme | 12-11 |
| Theme Attributes | 12-11 |
| Managing Themes and Subscriptions | 12-17 |

| | |
|---|-------|
| Understanding Theme Subscriptions | 12-18 |
| About Theme Subscriptions | 12-18 |
| About Modifying a Theme Subscription | 12-18 |
| Verifying if a Theme Refresh Is Needed | 12-19 |
| Refreshing a Subscribed Theme | 12-19 |
| Unsubscribing to a Theme | 12-20 |
| Re-subscribing to a Built-in Theme | 12-20 |
| Creating a Theme | 12-21 |
| Switching Themes | 12-22 |
| About Switching the Active Theme | 12-22 |
| Switching an Active Theme | 12-22 |
| Changing a Theme Identification Number | 12-23 |
| Copying a Theme | 12-24 |
| Deleting a Theme | 12-24 |
| Creating a Master Theme Application | 12-25 |
| Creating a Master Theme Application by Creating a New Application | 12-25 |
| Creating a Master Theme Application from an Existing Application | 12-25 |
| Managing Workspace Themes | 12-26 |
| Adding a Workspace Theme to the Theme Repository | 12-26 |
| Viewing an Existing Workspace Theme | 12-27 |
| Deleting a Workspace Theme | 12-27 |
| Modifying a Workspace Theme | 12-28 |
| Exporting a Workspace Theme | 12-28 |
| Managing Instance Themes | 12-28 |
| About Exporting and Importing Themes | 12-28 |
| Viewing Theme Reports | 12-29 |
| Viewing All Templates in a Theme | 12-29 |
| Viewing Theme Template Counts | 12-29 |
| Viewing File References | 12-30 |
| Viewing Class References | 12-30 |
| Viewing Template Substitution Strings | 12-31 |
| Creating Custom Themes | 12-32 |
| About Cascading Style Sheets | 12-32 |
| About Calling the JavaScript File from the Page Template | 12-33 |
| About Using Escaping Syntax in Substitution Strings | 12-33 |
| Selecting a Default Page Template | 12-34 |
| Selecting a Page-level Template Within a Theme | 12-34 |
| Selecting a Page-level Template for a Specific Page | 12-34 |
| Creating a New Template | 12-35 |
| Using Template Options | 12-35 |
| Managing Global Template Option Groups | 12-36 |

| | |
|--|-------|
| Managing Global Template Options | 12-37 |
| Managing Template Options | 12-39 |
| Viewing Template Reports | 12-42 |
| Managing Templates | 12-43 |
| Viewing Templates on the Templates Page | 12-43 |
| Viewing Templates Associated with a Specific Page | 12-44 |
| Replacing Templates | 12-44 |
| Replacing All Templates within an Application | 12-45 |
| Viewing Region Position Utilization by Page Template | 12-45 |
| Unsubscribing to Templates | 12-46 |
| Publishing Templates | 12-46 |
| Breadcrumb Templates | 12-46 |
| About Breadcrumb Style Navigation | 12-47 |
| Breadcrumb Template Attributes | 12-47 |
| Button Templates | 12-49 |
| Button Template Attributes | 12-49 |
| Legacy Calendar Templates | 12-51 |
| Supported Legacy Calendar Template Substitution Strings | 12-52 |
| Legacy Calendar Template Attributes | 12-52 |
| Label Templates | 12-53 |
| Label Template Attributes | 12-53 |
| About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help | 12-55 |
| List Templates | 12-56 |
| About Hierarchical Lists | 12-56 |
| List Template Attributes | 12-57 |
| Page Templates | 12-59 |
| Page Template Attributes | 12-59 |
| Required Dialog Page Template Attributes | 12-67 |
| Popup LOV Templates | 12-68 |
| Popup List of Values Template Attributes | 12-68 |
| Region Templates | 12-69 |
| Region Template Attributes | 12-70 |
| Report Templates | 12-72 |
| Generic Column Templates and Named Column Templates | 12-72 |
| Report Column Template Attributes for Generic Column Templates | 12-73 |
| Report Column Template Attributes for Named Column Templates | 12-77 |
| About Using JavaScript in Column Templates | 12-80 |
| Using Custom Cascading Style Sheets | 12-81 |
| About Uploading Cascading Style Sheets | 12-81 |
| Referencing an Uploaded Cascading Style Sheet in the Page Template | 12-81 |

13 Controlling Page Layout

| | |
|--|-------|
| About Page Layout in Oracle Application Express | 13-1 |
| Optimizing a Page for Printing | 13-2 |
| Selecting a Printer Friendly Template for an Application | 13-2 |
| Using f?p Syntax to Toggle to Printer Friendly Mode | 13-3 |
| Creating a Global Page to Display Components on Every Page | 13-3 |
| Creating a Global Page | 13-4 |
| Navigating to the Global Page | 13-4 |
| Managing Regions | 13-4 |
| About Regions | 13-5 |
| Creating a Region | 13-5 |
| Editing Regions | 13-6 |
| Editing Region Attributes | 13-7 |
| Controlling Region Positioning | 13-8 |
| Specifying a Region Header and Footer | 13-8 |
| Enabling Users to Customize a Page | 13-8 |
| Utilizing Region Caching in Page Designer | 13-9 |
| Specifying a Static Region ID | 13-10 |
| Adding a Region Image | 13-11 |
| Creating a Region Display Selector | 13-12 |
| Copying a Region | 13-13 |
| Deleting Regions | 13-13 |
| About Incorporating Content from Other Web Sites | 13-14 |
| About Managing Images | 13-15 |
| Rendering HTML Using Custom PL/SQL | 13-15 |

14 Managing Database Application Controls

| | |
|---|-------|
| Managing Page-Level Items | 14-1 |
| Viewing Page-Level Items | 14-2 |
| Creating Page-Level Items | 14-2 |
| Creating a Page-Level Item from the Rendering Tab | 14-3 |
| Adding a Page-Level Item from the Gallery | 14-3 |
| Creating a Static List of Values | 14-4 |
| Creating a Cascading List of Values | 14-8 |
| Creating a Shuttle Item on the Form Page in Page Designer | 14-10 |
| Editing Page-Level Items | 14-10 |
| Editing Page Item Attributes in the Property Editor | 14-11 |
| Defining Default Values for Page Items | 14-11 |
| Configuring Page Item Security | 14-12 |
| Creating a Quick Pick Selection | 14-14 |

| | |
|--|-------|
| Displaying Conditional Page Items | 14-15 |
| Displaying Read Only Page Items | 14-15 |
| Applying a Format Mask to an Item | 14-16 |
| Configuring Item Attributes to Warn Users of Unsaved Changes | 14-17 |
| Viewing Item Utilities | 14-18 |
| Understanding Page-Level Items | 14-18 |
| About the Differences Between Page Items and Application Items | 14-19 |
| About Item Naming Conventions | 14-19 |
| Referencing Item Values | 14-20 |
| About Referencing Items Using JavaScript | 14-21 |
| Working with Multiple Select List Item | 14-21 |
| Managing Dynamic Actions | 14-22 |
| About Dynamic Actions | 14-23 |
| About Dynamic Action Events | 14-24 |
| Creating a Dynamic Action | 14-26 |
| Editing Dynamic Actions | 14-29 |
| Defining Dynamic Action Event Scope | 14-30 |
| Deleting a Dynamic Action | 14-30 |
| About Calling JavaScript Using a Dynamic Action | 14-31 |
| Debugging Dynamic Actions | 14-31 |
| About Debugging Dynamic Actions | 14-31 |
| Debugging Dynamic Actions | 14-32 |
| Managing Buttons | 14-32 |
| What Actions Can a Button Perform? | 14-33 |
| Creating a Button | 14-33 |
| Editing a Button | 14-34 |
| Displaying a Button Conditionally | 14-35 |
| Configuring Button Attributes to Warn Users of Unsaved Changes | 14-35 |
| About Calling JavaScript from a Button | 14-36 |
| About the Relationship Between Button Names and REQUEST | 14-37 |
| About Branching with Buttons | 14-38 |
| Managing Trees | 14-38 |
| About Trees | 14-38 |
| Creating a Tree on New Page | 14-39 |
| Creating a Tree in Page Designer | 14-41 |
| Editing Tree Attributes | 14-44 |
| About Incorporating JavaScript into an Application | 14-44 |

15 Adding Navigation

| | |
|---|-------|
| About Using Lists as Navigation | 15-1 |
| Creating Tabs | 15-2 |
| About Tabs | 15-2 |
| About the Tabs Page | 15-3 |
| Checking for Tab Template Support | 15-4 |
| Creating a Tab | 15-5 |
| Editing Tabs | 15-7 |
| Editing Multiple Tabs Simultaneously | 15-7 |
| Managing Standard Tabs | 15-7 |
| Accessing Tab Reports | 15-8 |
| Creating Breadcrumbs | 15-8 |
| What Are Breadcrumbs? | 15-9 |
| Creating a Breadcrumb While Creating a Page | 15-9 |
| Creating a Breadcrumb Manually | 15-10 |
| About Creating a Breadcrumb Manually | 15-10 |
| Creating Breadcrumbs as Shared Components | 15-11 |
| Creating a Breadcrumb Region Using the Create Breadcrumb Wizard | 15-11 |
| Adding a Breadcrumb Region to a Page | 15-12 |
| Adding Entries to a Breadcrumb | 15-13 |
| About Creating Dynamic Breadcrumbs | 15-14 |
| Editing Breadcrumbs | 15-15 |
| Reparenting Breadcrumb Entries | 15-15 |
| Deleting Unused Breadcrumb Entries | 15-16 |
| Accessing Breadcrumb Reports | 15-16 |
| Creating Classic Navigation Bar | 15-17 |
| About Classic Navigation Bar Entries | 15-18 |
| Creating a Classic Navigation Bar Entry | 15-18 |
| Creating a Classic Navigation Bar Entry from Scratch | 15-18 |
| Creating a Classic Navigation Bar Entry for Feedback | 15-20 |
| Copying a Classic Navigation Bar Entry | 15-21 |
| Editing a Classic Navigation Bar Entry | 15-21 |
| Editing Multiple Classic Navigation Bar Entries Simultaneously | 15-22 |
| Accessing Classic Navigation Bar Entry Reports | 15-22 |
| Controlling Navigation Using Branches | 15-23 |
| Creating a Branch | 15-23 |
| About Branching Conditionally | 15-25 |

16 Managing Computations, Validations, and Processes

| | |
|---|-------|
| Understanding Page Computations | 16-1 |
| About Page Computations | 16-2 |
| Creating a Page Computation | 16-2 |
| Editing Page Computation Attributes | 16-3 |
| Editing the Computation Point and Source | 16-3 |
| Computation Points and Computation Syntax Example | 16-4 |
| Understanding Validations | 16-6 |
| About Validations | 16-6 |
| Creating a Validation | 16-7 |
| Editing a Validation | 16-8 |
| About Determining When Validations Execute | 16-9 |
| About Defining How Validation Error Messages Display | 16-9 |
| Processing Validations Conditionally | 16-10 |
| About Altering the Go to Error Link | 16-11 |
| What Happens When a Validation Fails? | 16-11 |
| Understanding Page Processes | 16-12 |
| About Page Processes | 16-12 |
| Creating Page Processes | 16-12 |
| Editing Page Process Attributes | 16-14 |
| Controlling When a Page Process Executes | 16-14 |
| Creating Page Process Success and Error Messages | 16-15 |
| Making a Page Process Conditional | 16-16 |
| About Branches | 16-16 |
| Understanding Application Processes | 16-16 |
| On Demand Application Processes | 16-17 |
| About Application Processes that Execute On New Instance | 16-17 |
| About Running an On Demand Process from a Page Request | 16-18 |
| Application Process Example | 16-19 |
| Creating an Application Process | 16-19 |
| Editing Application Process Attributes | 16-20 |
| Creating Application Process Error Messages | 16-20 |
| Viewing the Application Processes History Report | 16-21 |
| Understanding Application Computations | 16-22 |
| About Application Computations | 16-22 |
| About Application Computations that Execute On New Instance | 16-23 |
| Creating an Application Computation | 16-23 |
| Accessing the Application Computation History Report | 16-24 |
| Using the Attribute Dictionary | 16-24 |
| About the Attribute Dictionary | 16-25 |

17 Managing Shared Components

| | |
|--|-------|
| Accessing the Shared Components Page | 17-2 |
| Shared Components Page | 17-2 |
| Application Logic | 17-3 |
| Security | 17-4 |
| Other Components | 17-4 |
| Navigation | 17-5 |
| User Interface | 17-6 |
| Files | 17-6 |
| Data Sources | 17-7 |
| Reports | 17-7 |
| Globalization | 17-8 |
| Tasks Region | 17-8 |
| Workspace Objects Region | 17-9 |
| Managing Application Settings | 17-9 |
| Creating an Application Setting | 17-10 |
| Editing or Deleting Application Setting | 17-10 |
| Creating Lists of Values at the Application-Level | 17-11 |
| What is a List of Values? | 17-12 |
| Creating a Named LOV at the Application-Level | 17-12 |
| About Static LOVs | 17-12 |
| Editing an Existing LOV | 17-13 |
| Editing a LOV | 17-13 |
| Editing LOVs in Bulk | 17-13 |
| About Referencing Session State Within an LOV | 17-14 |
| Accessing LOV Reports | 17-14 |
| Managing Component Settings | 17-15 |
| Configuring Component Settings | 17-15 |
| Integrating with Google Calendar | 17-16 |
| Configuring Color Picker Display | 17-16 |
| Configuring Interactive Report Action Menu Structure | 17-17 |
| Dynamically Adding Sliders to Regions | 17-17 |
| Configuring Switch Defaults | 17-18 |
| Configuring Rich Text Editor Responsiveness | 17-18 |
| Viewing the Component Settings History Report | 17-19 |
| Using Shortcuts | 17-19 |
| What is a Shortcut? | 17-20 |
| Supported Shortcut Types | 17-20 |

| | |
|--|-------|
| Defining a Shortcut from Scratch | 17-21 |
| Copying an Existing Shortcuts | 17-22 |
| Editing Existing Shortcuts | 17-22 |
| Accessing Shortcut Reports | 17-23 |
| Managing Email Templates | 17-23 |
| Defining an Email Template | 17-24 |
| Viewing Sample Email Templates | 17-26 |
| About Including Dynamic Text | 17-27 |
| Editing Email Templates | 17-28 |
| Managing Static Application Files | 17-29 |
| Specifying the Location of Static Application Files | 17-29 |
| Uploading Static Application Files | 17-30 |
| Viewing the Static Application File Report | 17-30 |
| Referencing Static Application Files | 17-31 |
| Downloading Static Application Files | 17-31 |
| Downloading a Static Application File | 17-31 |
| Downloading All Static Application Files in a Zip | 17-32 |
| Deleting Static Application Files | 17-32 |
| Deleting a Static Application File | 17-32 |
| Deleting All Static Application Files | 17-33 |
| Managing Static Workspace Files | 17-33 |
| Uploading Static Workspace Files | 17-33 |
| Viewing Static Workspace File Report | 17-34 |
| About Referencing Static Workspace Files | 17-34 |
| Downloading Static Workspace Files | 17-35 |
| Downloading a Static Workspace File | 17-35 |
| Downloading All Static Workspace Files in a Zip | 17-35 |
| Deleting Static Workspace Files | 17-36 |
| Deleting a Static Workspace File | 17-36 |
| Deleting All Static Workspace Files | 17-36 |
| Managing Application-Level Items | 17-37 |
| Creating an Application-Level Item | 17-37 |
| Viewing Application Item Usage | 17-39 |
| Editing Application-Level Item Attributes | 17-39 |
| About Creating an Application-Level Item that Functions as a Global Variable | 17-40 |
| Creating Lists | 17-40 |
| About Static and Dynamic Lists | 17-41 |
| Creating Static Lists | 17-42 |
| About the Process of Creating a Static List | 17-42 |
| Creating a Static List from Scratch | 17-43 |
| About Adding Entries and Sublists to a Static List | 17-43 |

| | |
|---|-------|
| Adding Entries or Sublists from Scratch | 17-44 |
| Copying Static List Entries Between Lists | 17-46 |
| Reparenting Static List Entries | 17-46 |
| Managing Orphaned Static List Entries | 17-47 |
| Resequencing Static List Entries | 17-47 |
| Creating Dynamic Lists | 17-48 |
| About the Process of Creating a Dynamic List | 17-48 |
| Understanding Dynamic List Syntax | 17-48 |
| Creating a Dynamic List from Scratch | 17-50 |
| Copying a List | 17-51 |
| Copying a List from the Current Application | 17-51 |
| Copying a List Between Applications | 17-51 |
| Adding a List to a Page | 17-52 |
| Editing List Attributes | 17-53 |
| Accessing List Reports | 17-54 |
| Managing Navigation Menus | 17-55 |
| About Navigation Menus | 17-55 |
| Editing Navigation Menu Lists | 17-55 |
| Changing Where and How Navigation Menus Display | 17-56 |
| About Switching from Tabs to Navigation Menus | 17-57 |
| Managing Navigation Bar Lists | 17-57 |
| About Navigation Bars | 17-57 |
| Editing Navigation Bar Lists | 17-58 |
| Editing Navigation Bar User Interface Details | 17-58 |
| Managing Report Output | 17-59 |
| Printing a Report Region by Defining a Report Query | 17-59 |
| About Report Queries | 17-60 |
| Creating a Report Query | 17-60 |
| Editing a Report Query | 17-62 |
| Copying a Report Query | 17-62 |
| Formatting a Report Region or Report Query Using Report Layouts | 17-63 |
| About Report Layouts | 17-63 |
| About Report Layout Options | 17-64 |
| Creating a Report Layout | 17-64 |
| Editing a Report Layout | 17-65 |
| Copying a Report Layout | 17-65 |

18 Managing Application Data

| | |
|--|------|
| Creating Applications with Data Loading Capability | 18-1 |
| About Creating a Data Load Wizard | 18-2 |

| | |
|--|-------|
| Creating a Data Load Wizard | 18-4 |
| Re-creating Data Load Pages | 18-6 |
| Editing the Data Load Definition | 18-7 |
| About Using Collections | 18-8 |
| Managing REST Enabled SQL References | 18-9 |
| About REST Enabled SQL Service | 18-9 |
| Before You Begin: REST Enabled SQL Service Requirements | 18-10 |
| Creating a REST Enabled SQL Service Reference | 18-12 |
| Example: Creating a REST Enabled SQL Service Reference | 18-13 |
| Editing a REST Enabled SQL Service Reference | 18-15 |
| Deleting a REST Enabled SQL Service Reference | 18-17 |
| Creating or Updating Components to Use a REST Enabled SQL | 18-18 |
| Creating an Interactive or Classic Report Using a REST Enabled SQL Reference | 18-18 |
| Creating a Chart Using a REST Enabled SQL Reference | 18-20 |
| Creating a Calendar Using a REST Enabled SQL Reference | 18-23 |
| Specifying a REST Enabled SQL Reference in an Existing Component | 18-25 |
| Managing Web Source Modules | 18-27 |
| About Web Source Modules | 18-27 |
| Creating a Web Source Module | 18-28 |
| Editing or Deleting a Web Source Module | 18-30 |
| Copying a Web Source Module | 18-32 |
| Viewing Credential Utilization | 18-32 |
| Viewing Web Service Module History | 18-33 |
| Managing Remote Servers | 18-33 |
| About Remote Servers | 18-33 |
| Creating a Remote Server | 18-34 |
| Editing or Deleting a Remote Server | 18-35 |
| Viewing Remote Server Utilization | 18-36 |
| Viewing Remote Server History | 18-36 |
| Managing Web Credentials | 18-37 |
| About Credentials | 18-37 |
| Creating Web Credentials | 18-37 |
| Editing or Deleting Credentials | 18-38 |
| Viewing Credential Utilization | 18-39 |
| Viewing Credential History | 18-40 |
| Managing Legacy Web Services | 18-40 |
| About Web Services | 18-41 |
| Creating Web Service References | 18-42 |
| Accessing the Web Service References Page | 18-42 |
| Specifying an Application Proxy Server Address | 18-42 |

| | |
|--|-------|
| About Working with SSL Enabled Web Services | 18-43 |
| Creating Web Service References Based on a WSDL | 18-43 |
| About Creating Web Service References Based on a WSDL | 18-43 |
| Creating a Web Service Reference by Specifying a WSDL Document | 18-43 |
| Testing a Web Service Reference Created from a WSDL | 18-44 |
| Creating Web Service References Manually | 18-45 |
| About Creating Web Service References Manually | 18-45 |
| Creating a Web Service Reference Manually | 18-45 |
| Testing a Web Service Reference Created Manually | 18-46 |
| Creating RESTful Web Service References | 18-47 |
| About Creating RESTful Web Service References | 18-47 |
| Creating a RESTful Web Service Reference | 18-47 |
| Testing a REST Web Service Reference | 18-49 |
| Using the Web Service Reference Repository | 18-50 |
| Creating an Input Form and Report on a Web Service | 18-51 |
| About the Create Form and Report on Web Service Wizard | 18-51 |
| Creating a Form and Report by Adding a New Page | 18-51 |
| Creating a Form on a Web Service | 18-52 |
| About the Create Form on Web Service Wizard | 18-52 |
| Creating a Form by Adding a New Page | 18-53 |
| Creating a Report on a Web Service | 18-54 |
| How to Invoke a Web Service as a Process | 18-56 |
| Invoking a Web Service as a Process | 18-56 |
| Editing a Web Service Process | 18-58 |
| Viewing a Web Service Reference History | 18-59 |
| Accessing Data with Database Links | 18-59 |
| About Database Links | 18-60 |
| Creating a Database Link | 18-60 |
| Viewing an Existing Database Link | 18-60 |
| About DML Locking | 18-61 |
| About DML Locking | 18-61 |
| APEX_DML_LOCK_WAIT_TIME | 18-61 |
| FSP_DML_LOCK_ROW | 18-61 |

19 Extending Application Capabilities

| | |
|---|------|
| Sending Email from an Application | 19-1 |
| About Configuring Email | 19-2 |
| About Enabling Network Services | 19-2 |
| About Configuring Email in a Full Development Environment | 19-2 |
| About Email Provisioning | 19-3 |

| | |
|---|-------|
| About the Mail Queue | 19-3 |
| About the APEX_MAIL Package | 19-3 |
| Implementing Plug-ins | 19-4 |
| About Plug-ins | 19-4 |
| Accessing the Plug-ins Page | 19-5 |
| Creating a Plug-in | 19-5 |
| Editing a Plug-in | 19-7 |
| Adding Custom Attributes to a Plug-in | 19-7 |
| Uploading Files Associated with a Plug-in | 19-8 |
| Uploading a File to Associate with a Plug-in | 19-8 |
| Automatically Loading CSS and JavaScript Files | 19-9 |
| Adding Events to a Plug-in | 19-10 |
| Deleting a Plug-in | 19-10 |
| Viewing the Plug-in Repository | 19-11 |
| Importing a Plug-in from the Plug-in Page | 19-11 |
| Exporting a Plug-in from the Plug-in Page | 19-12 |
| Resetting the Plug-in Interactive Report | 19-13 |
| Viewing Plug-in Utilization Page | 19-13 |
| Viewing Plug-in History | 19-13 |
| Understanding jQuery and jQuery UI Support | 19-14 |
| About Upgrading to jQuery 3.1.1 | 19-14 |
| About Upgrading to jQuery UI 1.12.x | 19-14 |
| Manually Refreshing Oracle Application Express Components | 19-15 |

20 Managing Application Security

| | |
|---|-------|
| Understanding Administrator Security Best Practices | 20-1 |
| About Oracle Application Express Administrator Roles | 20-3 |
| About Configuring Oracle REST Data Services with Oracle Application Express | 20-3 |
| About Configuring Oracle HTTP Server with mod_plsql with Oracle Application Express | 20-4 |
| About Security Considerations When Using the Embedded PL/SQL Gateway | 20-5 |
| Managing Instance Security | 20-5 |
| About Creating Login Controls | 20-6 |
| About Enabling Public File Upload | 20-7 |
| About Restricting User Access by IP Address | 20-7 |
| About Specifying an Instance Proxy | 20-7 |
| About Utilizing Secure Sockets Layer | 20-8 |
| About Enabling RESTful Access | 20-8 |
| About Rejoin Sessions | 20-9 |
| About Isolating Workspaces | 20-9 |
| About Utilizing Session Timeout | 20-10 |

| | |
|---|-------|
| Restricting Password Reuse | 20-11 |
| About Using SQL in Websheets | 20-12 |
| About Enabling RESTful Services | 20-12 |
| Configuring Instance Settings | 20-13 |
| About Enabling Automation When Creating Workspaces | 20-13 |
| About Enabling and Configuring Email | 20-13 |
| About Configuring Storage Options | 20-14 |
| About Creating Wallets | 20-14 |
| About Configuring Workspace Purge Settings | 20-14 |
| Understanding Workspace Management | 20-15 |
| About Managing Workspace to Schema Assignments | 20-15 |
| About Managing Developers and Users | 20-15 |
| About Managing Component Availability | 20-16 |
| About Integrating with Oracle BI Publisher | 20-16 |
| About the Advantages of the Application Express Runtime Environment | 20-17 |
| Enabling Network Services in Oracle Database 11g or Later | 20-17 |
| When and Why Network Services Must be Enabled | 20-17 |
| Granting Connect Privileges Prior to Oracle Database 12c | 20-18 |
| Granting Connect Privileges in Oracle Database 12c or Later | 20-19 |
| Troubleshooting an Invalid ACL Error | 20-20 |
| Understanding Developer Security Best Practices | 20-21 |
| About Items of Type Password | 20-22 |
| Identifying At Risk Password Items | 20-23 |
| Viewing the Security Profiles Report | 20-23 |
| Viewing the Password Items Report | 20-23 |
| Understanding Cross-Site Scripting Protection | 20-24 |
| What Is Cross Site-scripting Security Breach? | 20-24 |
| Protecting HTML Regions and Other Static Areas | 20-24 |
| About Protecting Dynamic Output | 20-26 |
| About Protecting Report Regions | 20-27 |
| About Protecting Form Items | 20-28 |
| About Restricting Characters Entered on an Item | 20-28 |
| About Session State and Security | 20-29 |
| Preventing URL Tampering | 20-30 |
| How Session State Protection Works | 20-30 |
| Enabling Session State Protection | 20-30 |
| Configuring Session State Protection | 20-32 |
| About Securing File Uploads | 20-42 |
| Controlling Access to Applications, Pages, and Page Components | 20-42 |
| Understanding Access Control | 20-43 |
| About Adding Access Control | 20-43 |

| | |
|---|-------|
| About Access Control Authorization Schemes | 20-44 |
| About Configuring Access Control | 20-45 |
| About Exporting an Application with Access Control | 20-46 |
| Creating an Access Control Using the Create Application Wizard | 20-47 |
| Creating an Access Control Using the Create Page Wizard | 20-48 |
| Managing User Access | 20-49 |
| Configuring Access Control | 20-50 |
| About Controlling Access for Pages and Page Components | 20-50 |
| About Removing an Access Control Created with a Wizard | 20-51 |
| Managing Roles and User Assignments | 20-52 |
| About Application Access Control | 20-52 |
| Creating Access Control Roles | 20-53 |
| Editing or Deleting Access Control Roles | 20-54 |
| Adding User Role Assignments | 20-54 |
| Editing User Role Assignments | 20-55 |
| Establishing User Identity Through Authentication | 20-56 |
| Understanding Authentication | 20-56 |
| How Authentication Works | 20-56 |
| About Support for Deep Linking | 20-57 |
| About Determining Whether to Include Authentication | 20-57 |
| Creating an Authentication Scheme | 20-58 |
| Understanding Preconfigured Authentication Schemes | 20-59 |
| Application Express Accounts | 20-60 |
| Custom Authentication | 20-61 |
| Database Accounts | 20-62 |
| HTTP Header Variable | 20-63 |
| LDAP Directory Verification | 20-64 |
| No Authentication (using DAD) | 20-66 |
| Open Door Credentials | 20-67 |
| Oracle Application Server Single Sign-On Server | 20-67 |
| Social Sign-In | 20-68 |
| Managing Existing Authentication Schemes | 20-70 |
| Viewing the Authentication Scheme Associated with an Application | 20-70 |
| Changing the Authentication Scheme Associated with an Application | 20-71 |
| Creating a Login Page | 20-71 |
| Viewing Authentication Scheme Reports | 20-72 |
| Providing Security Through Authorization | 20-72 |
| How Authorization Schemes Work | 20-72 |
| Creating and Editing an Authorization Scheme | 20-73 |
| About Authorization Scheme Types | 20-73 |
| Creating an Authorization Scheme | 20-74 |

| | |
|--|-------|
| Editing Attributes of an Existing Authorization Scheme | 20-75 |
| Changing the Evaluation Point Attribute | 20-75 |
| About Resetting Authorization Scheme State | 20-76 |
| Attaching an Authorization Scheme to an Application, Page, or Components | 20-76 |
| Attaching an Authorization Scheme to an Application | 20-77 |
| Attaching an Authorization Scheme to a Page | 20-77 |
| Attaching an Authorization Scheme to a Control or Component | 20-78 |
| Viewing Authorization Reports | 20-78 |

21 Managing Application Globalization

| | |
|---|-------|
| About Apply Format Masks to Items | 21-2 |
| Understanding Application Translation and Globalization Support | 21-2 |
| About the Translation Process | 21-2 |
| About Language Identification | 21-3 |
| About the Rules for Translating Database Applications | 21-3 |
| How Translated Applications Are Rendered | 21-4 |
| Making Application Attributes Translatable | 21-4 |
| About Shortcuts that Support Translatable Messages | 21-4 |
| About Messages | 21-5 |
| About Dynamic Translation Text Strings | 21-5 |
| About Translating Region Titles | 21-5 |
| About Translating Templates | 21-6 |
| Specifying the Primary Language for an Application | 21-7 |
| About Translating Applications for Multibyte Languages | 21-8 |
| Understanding the Translation Process | 21-8 |
| Step 1: Define Application Language | 21-9 |
| Step 2: Seed and Download to a Translation File | 21-9 |
| Seeding Translatable Text | 21-10 |
| Downloading an XLIFF File | 21-10 |
| Step 3: Translate the XLIFF File | 21-12 |
| Step 4: Upload and Apply a Translated XLIFF Document | 21-13 |
| Uploading a Translated XLIFF Document | 21-13 |
| Applying an Uploaded XLIFF Document | 21-13 |
| Deleting an Uploaded XLIFF Document | 21-14 |
| Step 5: Publish the Application | 21-15 |
| Manually Editing a Translation | 21-15 |
| Translating Messages | 21-16 |
| About Translating Messages | 21-17 |
| Translating Messages Used in PL/SQL Procedures | 21-17 |
| About Translating Messages Used in PL/SQL Procedures | 21-17 |

| | |
|--|-------|
| Defining a Translation Message | 21-17 |
| Interactive Grid Messages Requiring Translation | 21-18 |
| Translating Messages Used Internally by Oracle Application Express | 21-38 |
| Creating Text Messages | 21-38 |
| Editing Text Messages | 21-39 |
| Internal Messages Requiring Translation | 21-39 |
| Interactive Report Messages Requiring Translation | 21-44 |
| Translating Data That Supports List of Values | 21-61 |
| Defining a Dynamic Translation | 21-62 |
| Understanding Supported Globalization Codes | 21-62 |
| Oracle Application Express Globalization Codes | 21-62 |

22 Managing Application Performance

| | |
|--|-------|
| About Best Practices to Improve Performance | 22-1 |
| Identifying Performance Issues | 22-2 |
| About Troubleshooting Slow Running Applications | 22-2 |
| Diagnosing Performance Issues | 22-3 |
| About Debugging Problematic SQL Queries | 22-3 |
| About Addressing Slow Queries | 22-3 |
| Utilizing Logs and Reports | 22-3 |
| How to Manage Log Files | 22-4 |
| Viewing the Application Dashboard Report | 22-5 |
| Workspace Activity Reports | 22-5 |
| Instance Activity Reports | 22-6 |
| About Utilizing Database Reporting | 22-6 |
| About Database Parameters that Impact Performance | 22-7 |
| About Limiting Resources | 22-8 |
| About Uploading Static Files to Your Web Server | 22-8 |
| Creating Custom Activity Reports Using APEX_ACTIVITY_LOG | 22-9 |
| Enabling Logging for an Application | 22-9 |
| APEX_ACTIVITY_LOG | 22-10 |
| APEX_ACTIVITY_LOG Sample Query | 22-11 |

23 Debugging an Application

| | |
|--|------|
| Running Advisor to Check Application Integrity | 23-1 |
| About Oracle Application Express Advisor | 23-2 |
| Running Advisor on an Entire Application | 23-2 |
| Running Advisor on a Single Page | 23-3 |
| Reviewing Session State | 23-3 |

| | |
|---|------|
| About Monitoring Application and Page Resource Use | 23-3 |
| Enabling SQL Tracing and Using TKPROF | 23-4 |
| Utilizing Debug Mode | 23-4 |
| About Debug Mode | 23-5 |
| Enabling and Disabling Debug Mode for an Application | 23-5 |
| Running an Application in Debug Mode | 23-6 |
| Viewing Debug Reports in Development Mode | 23-6 |
| About Using f?p Syntax to Access Debug Mode | 23-7 |
| Viewing Debug Reports from a Running Application | 23-7 |
| Viewing Debug Reports | 23-8 |
| About Removing Controls and Components to Isolate a Problem | 23-9 |

24 Deploying an Application

| | |
|---|-------|
| About Suggested Environments | 24-2 |
| System Development Life Cycle Methodologies to Consider | 24-2 |
| About Iterative Vs Planned Development | 24-3 |
| About the Advantages of Creating Prototypes | 24-3 |
| Waterfall | 24-3 |
| Spiral | 24-4 |
| Rapid Application Development | 24-4 |
| Understanding the Deployment Process | 24-4 |
| About Moving an Application to Another Instance | 24-4 |
| Deployment Options to Consider | 24-5 |
| How to Move an Application to Another Instance | 24-6 |
| About Managing Database Objects | 24-6 |
| How to Create a Custom Application | 24-7 |
| How Creating a Custom Application Simplifies Deployment | 24-8 |
| Creating a Custom Application | 24-9 |
| Accessing the Supporting Objects Utility | 24-9 |
| Supporting Objects Page | 24-9 |
| Adding an Access Control List to a Custom Application | 24-12 |
| Installing Supporting Objects | 24-13 |
| Deleting Supporting Objects Scripts, Messages, and Installation Options | 24-13 |
| Upgrading a Custom Application | 24-14 |
| Defining an Upgrade Script | 24-14 |
| Upgrading a Custom Application | 24-15 |
| Deinstalling Supporting Objects | 24-15 |
| Viewing an Install Summary | 24-16 |
| Using Build Options to Control Configuration | 24-16 |
| About Build Options | 24-17 |

| | |
|---|-------|
| Creating Build Options | 24-18 |
| Including or Excluding Build Options | 24-18 |
| Selecting a Build Option | 24-19 |
| Deleting Build Options and Associated Components | 24-19 |
| Removing Components Associated with a Build Options | 24-20 |
| Deleting Build Options Permanently | 24-20 |
| Viewing Build Option Reports | 24-21 |
| Viewing Build Option Utilization | 24-21 |
| Viewing Build Option History | 24-21 |
| Exporting Build Options or Build Option Status | 24-22 |
| About Publishing the Database Application URL | 24-23 |
| About Publishing the Websheet Application URL | 24-24 |
| Exporting an Application and Application Components | 24-25 |
| About the Export Process | 24-25 |
| About The Export Page | 24-26 |
| Exporting an Application | 24-27 |
| Exporting Application Components | 24-30 |
| Exporting a Workspace | 24-31 |
| Exporting a Page in an Application | 24-32 |
| Exporting a Websheet | 24-33 |
| About Exporting Static Files | 24-34 |
| Exporting Themes | 24-34 |
| Exporting Plug-ins | 24-34 |
| Exporting User Interface Defaults | 24-35 |
| About Exporting User Interface Defaults | 24-35 |
| Exporting User Interface Defaults from the Export Page | 24-36 |
| Exporting User Interface Defaults from the User Interface Defaults Page | 24-36 |
| Exporting Team Development Feedback | 24-37 |
| Exporting Script Files from the Script Repository | 24-38 |
| Importing Export Files | 24-38 |
| Importing an Application, Page or Component Export | 24-39 |
| About Importing Application Groups | 24-40 |
| Importing a Websheet | 24-40 |
| Importing Plug-ins | 24-41 |
| Importing Themes | 24-41 |
| Importing User Interface Defaults | 24-42 |
| Importing Team Development Feedback | 24-43 |
| Installing Export Files | 24-43 |
| About Installing Export Files | 24-43 |
| Accessing the Export Repository | 24-44 |
| Installing an Application Export from the Export Repository | 24-44 |

| | |
|---|-------|
| Installing Other Files from the Export Repository | 24-45 |
| Deleting Files from the Export Repository | 24-46 |
| Installing Export Files from SQL*Plus | 24-46 |
| Using SQL*Plus to Install Export Files | 24-46 |

25 Managing Application Legacy Components

| | |
|---|-------|
| Managing Legacy Charts | 25-1 |
| About Chart Legacy Types | 25-2 |
| About Legacy HTML5 Charts | 25-2 |
| About Legacy Flash Charts | 25-2 |
| Creating SQL Queries for Legacy Charts | 25-2 |
| Legacy Chart Syntax | 25-3 |
| Dial Chart Syntax (Flash and HTML5) | 25-3 |
| Legacy Multiple Series Syntax (Flash and HTML5) | 25-3 |
| Legacy Range Chart Syntax (Flash and HTML5) | 25-4 |
| Legacy Scatter Chart Syntax (Flash and HTML5) | 25-4 |
| Legacy Candlestick Chart Syntax (Flash and HTML5) | 25-4 |
| Legacy Gantt Chart Syntax (Flash Only) | 25-4 |
| Adding a Legacy Chart to a New Page | 25-5 |
| Adding a Legacy Chart in Page Designer | 25-7 |
| Editing Legacy Charts | 25-8 |
| Editing Chart Attributes | 25-8 |
| Switching an HTML5 Chart to a Flash Chart | 25-9 |
| Enabling Asynchronous Updates | 25-10 |
| About Saving Flash Charts | 25-11 |
| Using Custom XML with Flash and HTML5 Charts | 25-12 |
| Creating a Legacy Master Detail Form | 25-13 |
| Managing Legacy Tabular Forms | 25-16 |
| About Creating Tabular Forms | 25-16 |
| About Running a Tabular Form | 25-17 |
| How Tabular Forms Work | 25-18 |
| Creating a Legacy Tabular Form | 25-19 |
| Managing Legacy Calendars | 25-20 |
| Editing Calendar Attributes in the Property Editor | 25-20 |
| About Supported Calendar Substitution Strings in Legacy Calendars | 25-21 |
| Viewing Legacy Components in Gallery | 25-21 |

| | | |
|----------|---|------|
| A | About Item Types | |
| | Available Item Types | A-1 |
| B | Oracle Application Express Limits | |
| | Component Limits | B-1 |
| C | Available Conditions | |
| | What Are Conditions? | C-1 |
| | Conditions Available in App Builder | C-1 |
| D | About Granted Privileges | |
| | About Granting Privileges to Database Users | D-1 |
| | Understanding Privileges Granted to PUBLIC | D-1 |
| | Packages | D-2 |
| | Procedures | D-4 |
| | Functions | D-4 |
| | Tables | D-5 |
| | Views | D-5 |
| | Sequences | D-10 |
| | Types | D-10 |

Index

Preface

Oracle Application Express App Builder User's Guide describes how to use the Oracle Application Express development environment to build and deploy database-centric web applications. Oracle Application Express turns a single Oracle database into a shared service by enabling multiple workgroups to build and access applications as if they were running in separate databases.

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

Oracle Application Express App Builder User's Guide is intended for application developers who are building database-centric web applications using Oracle Application Express. The guide describes how to use the Oracle Application Express development environment to build, debug, manage, and deploy applications.

To use this guide, you must have a general understanding of relational database concepts and an understanding of the operating system environment under which you run Oracle Application Express.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see these Oracle resources:

- *Oracle Application Express Release Notes*
- *Oracle Application Express Installation Guide*

- *Oracle Application Express End User's Guide*
- *Oracle Application Express Administration Guide*
- *Oracle Application Express SQL Workshop Guide*
- *Oracle Application Express API Reference*
- *Oracle Application Express Application Migration Guide*
- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*
- *Oracle Database SQL Language Reference*
- *SQL*Plus User's Guide and Reference*

Conventions

The following text conventions are used in this document:

| Convention | Meaning |
|-----------------|--|
| boldface | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| <i>italic</i> | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| monospace | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

Changes in This Release

This preface contains:

- [Changes in Oracle Application Express Release 18.2](#)
- [Changes in Oracle Application Express Release 18.1](#)

Changes in Oracle Application Express Release 18.2

The following are changes in *Oracle Application Express App Builder User's Guide* for release 18.2.

- [New Features](#)
- [Deprecated and Desupported Features](#)
- [Other Changes](#)

New Features

The following features are new in Oracle Application Express release 18.2:

- Declarative Static List of Values (LOVs)
 - Page Designer now includes a declarative approach to creating static list of values (LOVs).
See "[Creating a Static List of Values.](#)"
- Productivity and Sample Apps
 - **Packaged apps** have been renamed **productivity and sample apps**.
 - On the Workspace home page, the **Packaged Apps** icon has been renamed the **App Gallery** icon.
 - The **Packaged Apps** page is now called the **App Gallery** page.
See "[Utilizing the App Gallery.](#)"
- Master Detail Forms
 - Side by Side Master Detail
A new **Side by Side** master detail form creates a single page master detail utilizing a side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report(s).
 - Existing Master Detail Forms Renamed
Single Page master detail has been renamed **Stacked** master detail. **Two Page** master detail is now called **Drill Down** master detail.

- See "[Managing Master Detail Forms.](#)"
- Create Page Wizard
 - **Master Detail** has moved up one level and now includes three options: **Stacked**, **Side by Side**, and **Drill Down**.
 - **Dashboard** enables the creation of a first-cut dashboard based on sample data that you can later customize using Page Designer. Choose from a number of different chart layouts as the starting point for your page.
- See "[Available Component Page Types](#)"
- Create Application Wizard
 - New Advanced Setting, **Add "Built with APEX" to Footer**. When set to **Yes**, Oracle Application Express adds the text "Built with Love using Oracle APEX" to the footer of every page.
 - See "[Available Settings in the Create Application Wizard.](#)"
 - Side by Side Master Detail. This option creates a single page master detail utilizing side by side layout and report regions with modal edit windows.
 - See "[Managing Master Detail Forms.](#)" and "[Creating a Master Detail Form Using the Create Application Wizard.](#)"

Deprecated and Desupported Features

See "Deprecated Features" and "Desupported Features" in *Oracle Application Express Release Notes*.

Other Changes

The following are additional changes in Oracle Application Express release 18.2:

- All content has been updated to reflect new functionality. Desupported functionality has been removed.
- Screen captures and graphics have been added and updated to reflect current release user interface enhancement.

Changes in Oracle Application Express Release 18.1

The following are changes in *Oracle Application Express App Builder User's Guide* for release 18.1.

- [New Features](#)
- [Deprecated and Desupported Features](#)
- [Other Changes](#)

New Features

The following features are new in Oracle Application Express release 18.1:

- REST Enabled SQL Support

Use REST Enabled SQL references to execute SQL queries or PL/SQL defined at the component-level on a remote database. Create or update classic reports, interactive reports, CSS calendars, or charts to point to data on a remote database using a REST Enabled SQL reference.

See "[Managing REST Enabled SQL References.](#)"

- Web Source Modules

Create and maintain Web Source Modules within Shared Components that function as references to REST services.

See "[Managing Web Source Modules.](#)"

- New Create App Wizard

- Add application-level **features** to your application. Available features include: About Page, Access Control, Activity Reporting, Configuration Options, Feedback, and Theme Selection.

- Customize user interface options such as Theme Style, application icon, and page icons.

- Select lookup display values for foreign-key based columns.

- Select drill down report columns to declaratively link data in one report with another.

- Application blueprints: View an application blueprint and directly edit the application definition in JSON format.

- Modify application attribute defaults: Control labels and form controls by pointing and clicking or using JSON application defaults.

See "[Creating Database Applications](#)"

- Enhanced Create Page Wizard

- Add application-level **features** to your application. Available features include: About Page, Access Control, Activity Reporting, Configuration Options, Feedback, Job Reporting, Login Page, and Theme Selection.

See "[Managing Pages in a Database Application.](#)"

- Global Search

- Provides quick navigation and a unified search experience across Oracle Application Express.

- Activate by clicking Search from header or by using the keyboard shortcut Ctrl + Quote (') (US Keyboards only)

- Navigate to most screens within App Builder and quickly jump across applications and pages.

- Search in Page Designer by focussing on page components.

See "[Global Search.](#)"

- Page Designer Improvements

- Updated user interface to improve usability.

- When an attribute has focus, the group becomes highlighted.

- Quickly access a group, by clicking **Go to Group** and selecting the group. To return the default display, click **Go to Group** again and select **Expand All**.

- Click **Pin Filter** on the right side of the search field to have search terms you enter remain persistent (or “sticky”) as you click around Page Designer. Click **Pin Filter** again to disable it.

See "[Right Pane of Page Designer \(Property Editor\)](#)."

- Developer Toolbar Enhancements

- If a page has one or more JavaScript errors, a red error button displays on the left side of the Runtime Developer toolbar. If you have enabled **Auto Hide**, the Runtime Developer toolbar displays indicating the error.
- New Page Info option includes two options:
 - * **Show Layout Columns** toggles between **Show Layout Columns** and **Hide Layout Columns**.
 - * **Show Page Timing** displays the Page Performance Timing dialog. Click **Copy** to copy the data in table form and then paste it into another application. Click **Clear** to remove the current timing events.

See "[Runtime Developer Toolbar](#)."

- Universal Theme and User Interface Enhancements

- Includes Font APEX 2 which features two sets of icons: Small and Large.
- UI Optimizations:
 - * New Top Navigation Tabs template that can be used when the navigation position is set to top. Provides a simplified tabs-like UI and on small screens is automatically positioned near the bottom of the screen.
 - * Standard Region and Carousel Region templates now support placing an icon in the region header.
- General Enhancements
 - * Tweaked and polished several templates, template options, and styles.
 - * Required asterisk for form items is now consistent across label templates.
 - * Carousel regions now support touch swipes.
 - * Cards have a new "Block" template option with more color.

- General Mobile Improvements

- Universal Theme has been optimized for mobile devices. Desktop UI now supports all mobile friendly components such as List View region, Column Toggle Report region, and Reflow Report.
- Support touch for jQuery UI mouse interactions including draggable, droppable, and sortable. This can be seen in various places such as in Page Designer drag and drop and adjusting splitter positions.
- Expose touch events through Dynamic Actions for all apps.

- Advisor Improvements

- New Accessibility checks to check your apps for common accessibility issues.
- New Performance check for inclusion of compatibility JavaScript.

See "[Running Advisor to Check Application Integrity](#)" and "Testing Apps for Accessibility" in *Oracle Application Express Accessibility Guide*.

- Social Sign-In

- Social Sign-In preconfigured authentication scheme supports authentication with Google, Facebook, and other social network that supports OpenID Connect or OAuth2 standards.
- New authentication scheme attribute "Switch in Session" for schemes that can be set in the current session.

See "[Establishing User Identity Through Authentication](#)."

Deprecated and Desupported Features

See "Deprecated Features" and "Desupported Features" in *Oracle Application Express Release Notes*.

Other Changes

The following are additional changes in Oracle Application Express release 18.1:

- All content has been updated to reflect new functionality. Desupported functionality has been removed.
- Screen captures and graphics have been added and updated to reflect current release user interface enhancement.
- In this publication, legacy application components are grouped together. Legacy application components are components Oracle still supports but does not recommend using.

See "[Managing Application Legacy Components](#)."

1

Quick Start

"Quick Start" offers a quick introduction to Oracle Application Express. This section explains how to access your development environment, describes the Workspace home page, the App Builder home page, and the Application home page, explains how to edit your account profile and user preferences, and how to access product documentation.

Tip:

If you are accessing an on-premises installation, complete the installation process before continuing. See *Oracle Application Express Installation Guide*.

- [About Accessing Your Development Environment](#)
How you sign in and access Oracle Application Express depends upon your user role and where Oracle Application Express resides. Oracle Application Express may reside in a local on-premises Oracle Database or in a hosted environment, such as a Oracle Cloud service. The login credentials you use to sign in differ depending upon the installation type.
- [Understanding Oracle Application Express](#)
Oracle Application Express is a hosted declarative development environment for developing and deploying database-centric web applications.
- [Understanding the Workspace Home Page](#)
When you sign in to Oracle Application Express, the Workspace home page appears. Use the Workspace home page to access key development components including App Builder, SQL Workshop, Team Development, and the App Gallery.
- [Understanding App Builder](#)
Developers use App Builder to create and manage applications and application pages.
- [Changing Your Profile or Password](#)
Edit your account profile to update your email address, edit your first or last name, upload a profile photo, or change your password.
- [Editing User Preferences](#)
Edit your user preferences to update the default workspace schema, specify a default date format, and control how Oracle Application Express manages windows and tabs when running applications.
- [Using Oracle Application Express Documentation](#)
Oracle Application Express provides a hosted online documentation library and field-level Help. You access, search, or download the online documentation library by going to the Oracle Help Center.

About Accessing Your Development Environment

How you sign in and access Oracle Application Express depends upon your user role and where Oracle Application Express resides. Oracle Application Express may reside in a local on-premises Oracle Database or in a hosted environment, such as a Oracle Cloud service. The login credentials you use to sign in differ depending upon the installation type.

Oracle Application Express On-premises

If you are running Oracle Application Express on-premises, sign in to your Oracle Application Express workspace.

See Also:

"About Accessing Oracle Application Express" and "Signing In When Workspace Requests Is Enabled" in *Oracle Application Express Installation Guide*.

Oracle Database Cloud

Oracle Application Express is installed and enabled in Oracle Database Cloud services such as Oracle Database Exadata Express Cloud Service and Oracle Database Cloud - Database Schema Service.

Oracle Application Express is available in Exadata Cloud Service and Database Cloud Service. However, you need to manually customize your databases to install and enable Oracle Application Express.

See Also:

"Using Oracle Database Exadata Express Cloud Service" and "Using Oracle Database Cloud - Database Schema Service"

Understanding Oracle Application Express

Oracle Application Express is a hosted declarative development environment for developing and deploying database-centric web applications.

When you install Oracle Application Express, you can install either a runtime environment or a full development environment.

- [What is Oracle Application Express?](#)
- [How Oracle Application Express Works](#)
- [About the Differences Between Runtime and Full Development Environments](#)

What is Oracle Application Express?

Oracle Application Express is a hosted declarative development environment for developing and deploying database-centric web applications. Thanks to built-in features such as user interface themes, navigational controls, form handlers, and flexible reports, Oracle Application Express accelerates the application development process.

The Application Express engine renders applications in real time from data stored in database tables. When you create or extend an application, Oracle Application Express creates or modifies metadata stored in database tables. When the application is run, the Application Express engine then reads the metadata and displays the application.

To provide stateful behavior within an application, Oracle Application Express transparently manages session state in the database. Application developers can get and set session state using simple substitutions and standard SQL bind variable syntax.

How Oracle Application Express Works

Oracle Application Express installs with your Oracle database and consists of data in tables and PL/SQL code. Whether you run the Oracle Application Express development environment or run an application built using Oracle Application Express, the process is the same. Your browser sends a URL request that is translated into the appropriate Oracle Application Express PL/SQL call. After the database processes the PL/SQL, the results are relayed back to your browser as HTML. This cycle happens each time you either request or submit a page.

Oracle Application Express does not use a dedicated database connection. Instead, each request is made through a new database session, consuming minimal CPU resources. Application session state is managed in the database tables by the Application Express engine.

Behind the scenes, the Application Express engine renders and processes pages. The Application Express engine also performs these tasks:

- Session state management
- Authentication services
- Authorization services
- Page flow control
- Validation processing

About the Differences Between Runtime and Full Development Environments

When you install Oracle Application Express, you can install two different environments:

- **Runtime environment.** In a runtime environment users can run applications but cannot modify them. Select this option for production implementations.

- **Full development environment.** In a full development environment, users can develop, modify, run, and delete Application Express applications. This option provides complete access to the Oracle Application Express environment described in this document.

 **See Also:**

- *Oracle Application Express Installation Guide*
- "Installing Exported Applications into a Runtime Environment" in *Oracle Application Express Administration Guide*

Understanding the Workspace Home Page

When you sign in to Oracle Application Express, the Workspace home page appears. Use the Workspace home page to access key development components including App Builder, SQL Workshop, Team Development, and the App Gallery.

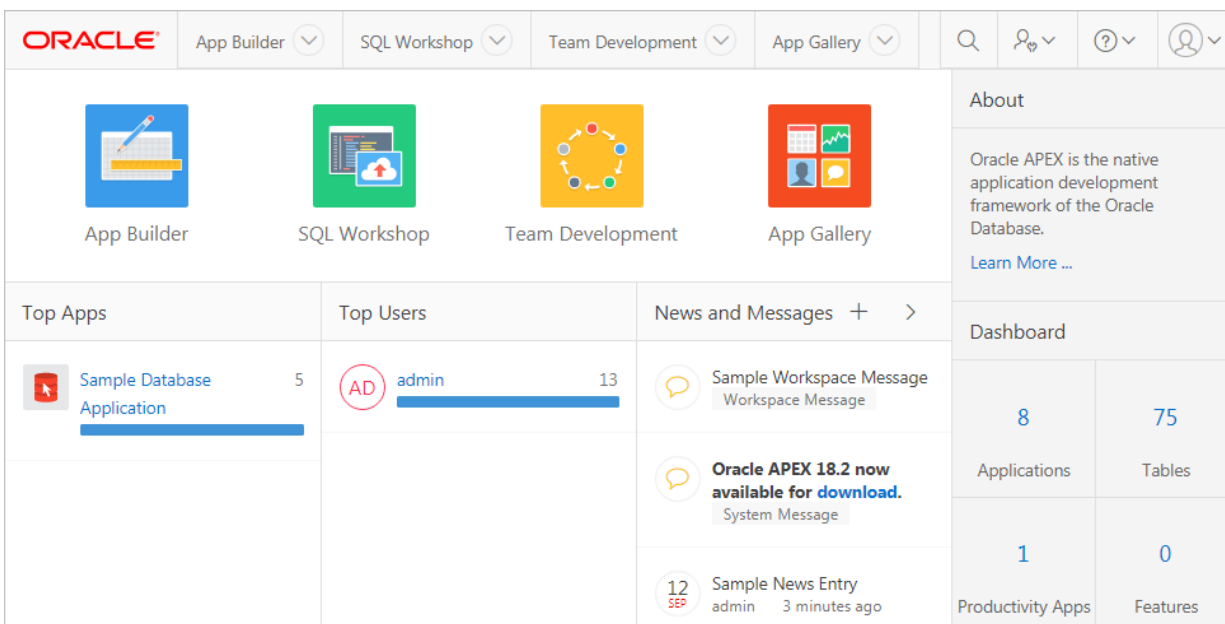
Each Oracle Application Express development instance can contain multiple workspaces. A **workspace** is a virtual private database which enables multiple users to work within the same Oracle Application Express installation while keeping their objects, data and applications private. This section describes the sections that make up the Workspace home page.

- [Workspace Home Page](#)
- [Top Apps and Top Users Dashboards](#)
- [News and Messages Region](#)
- [Developer Navigation Tools](#)
- [Available Updates Region](#)

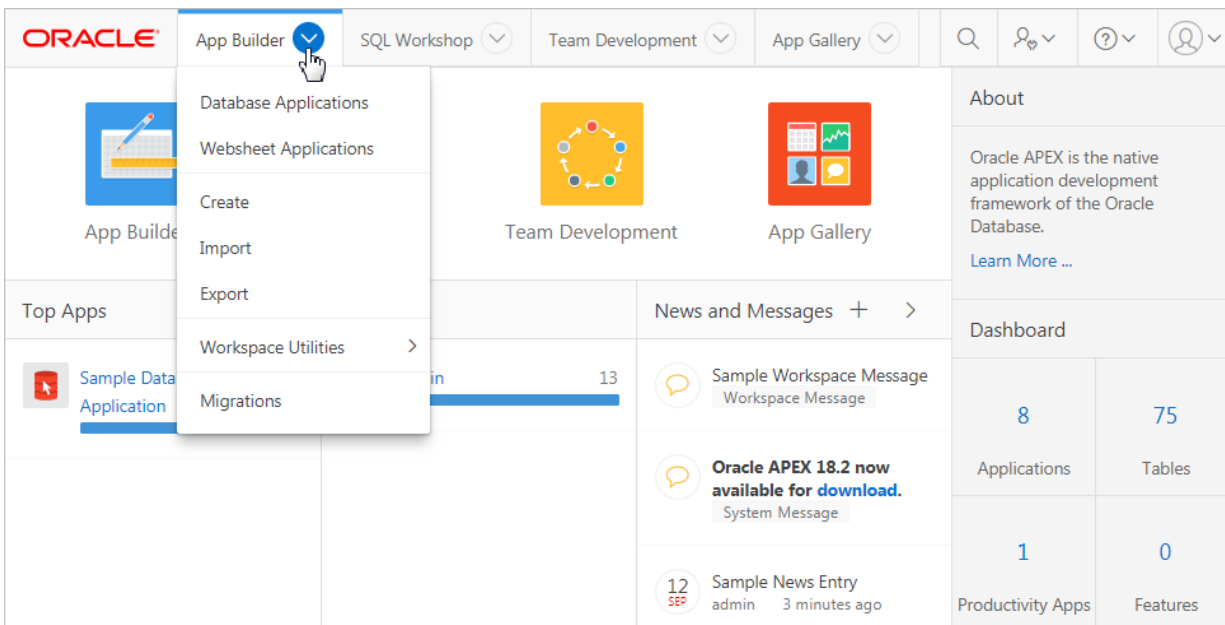
Workspace Home Page

The center of the Workspace home page features the following large icons:

- **App Builder** - Use App Builder to create an application, composed of a set of HTML pages, based on database objects.
- **SQL Workshop** - Use the SQL Workshop to access tools for viewing and managing database objects.
- **Team Development** - Use Team Development to track new features, non-feature related tasks (or To Do tasks), bugs, and milestones. Users can also provide real-time feedback which then can be categorized into features, general tasks, or bugs.
- **App Gallery** - Links to the App Gallery page. Install and run productivity or sample applications to learn more about the types of applications you can build and how to construct specific types of pages.



A menu bar also displays at the top of the page. Click a menu to access an alternate navigation path to different components within App Builder, SQL Workshop, Team Development, and App Gallery. The illustration that follows shows the **App Builder** menu.



 **See Also:**

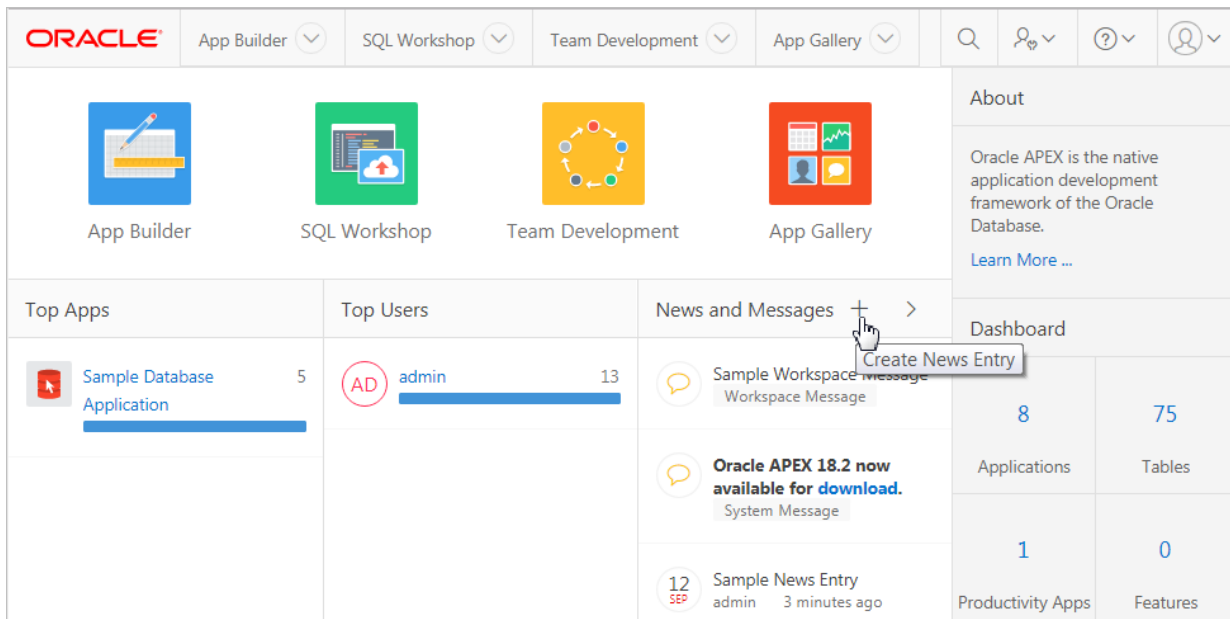
- " App Builder Concepts"
- *Oracle Application Express SQL Workshop Guide*
- "Managing the Application Life Cycle with Team Development"
- "Utilizing the App Gallery"

Top Apps and Top Users Dashboards

The Top Apps and Top Users regions display at the bottom of the page. These regions offer real time information about development activities in the current workspace.

News and Messages Region

The News and Messages region displays to the right of Top Apps and Top Users.



The screenshot shows the Oracle APEX Workspace Home Page. At the top, there is a navigation bar with the Oracle logo and several tabs: App Builder, SQL Workshop, Team Development, and App Gallery. Below the navigation bar, there are four main icons: App Builder, SQL Workshop, Team Development, and App Gallery. The main content area is divided into three sections: Top Apps, Top Users, and News and Messages. The Top Apps section shows a bar chart for 'Sample Database Application' with a value of 5. The Top Users section shows a bar chart for 'admin' with a value of 13. The News and Messages section displays three messages: a 'Sample Workspace Message', a 'System Message' about 'Oracle APEX 18.2 now available for download', and a 'Sample News Entry' by 'admin' from 3 minutes ago. A 'Create News Entry' button is visible in the News and Messages section. On the right side, there is a 'Dashboard' section with a table showing 'Applications' (8) and 'Tables' (75), and 'Productivity Apps' (1) and 'Features' (0).

The News and Messages region can display three types of messages:

- News entries
- System messages
- Workspace Announcements

News entries include a date to the left of the message text. Developers can create news entries to communicate with other workspace users. To add a news entry, click the **Create News Entry** icon which resembles a plus sign (+). To edit existing news entries, click **Edit News Entry** which resembles a greater than sign (>).

Workspace announcements and **system messages** include a gold icon to the left of message text. Depending upon the message type, a textual identifier, either *System Message* Or *Workspace Announcement*, also displays beneath the message text.

 **See Also:**

- ["Managing News Entries"](#)
- "Defining a System Message" and "Defining a Workspace Announcement" in *Oracle Application Express Administration Guide*

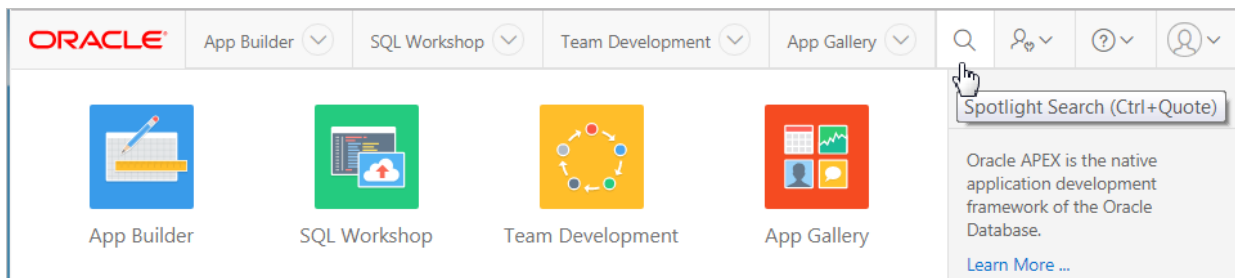
Developer Navigation Tools

Developer Navigation tools display in the header region on many Oracle Application Express pages, including the Workspace home page, App Builder home page, the Application home page, Page Designer, and Shared Components.

- [Global Search](#)
- [Administration Menu](#)
- [Help Menu](#)
- [Account Menu](#)

Global Search

The Global Search field displays in the header region in upper right corner of Application Express header page and provides an effective way to navigate App Builder.



Key features of Global Search include:

- Offers effortless navigation within a workspace enabling you to quickly find applications, pages, Shared Components, Application Express menus, and so on.
- Search for component names and immediately set focus to those items in Page Designer.
- Provides a unified search experience throughout Application Express.

About Using Global Search

You can activate Global Search in two ways:

- Click the **Spotlight Search** icon in the upper right corner
- Use the keyboard shortcut (Ctrl + Quote) (') (US Keyboards only)

Use Global Search to search all applications in the workspace no matter what your location in the current workspace. For example, if your workspace contains the *Sample Database Application*, entering `Sample` will find it. Similarly, you can access global navigation menus that appear at the top window adjacent to the Oracle logo. For example, enter `SQL Commands` in the Global Search field to instant access SQL Commands.

To close the Global Search dialog, press the `Esc` key, or click anywhere outside the results.

Syntax to Navigate to a Page or Search for a Regular Expression

Within App Builder and on the Application home page, use the following syntax to navigate to a specific page or search for a regular expression:

```
page_ID  
application_ID:page_ID  
application_ID-page_ID  
regexp:expression
```

Where:

- `application_ID` is the numeric identification number (or ID) that identifies an application.
- `page_ID` is the numeric identification number that identifies a specific page.
- `regexp:` is literal text.
- `expression` is a regular expression to search for.

The following example navigates you to page 10 of application 570:

```
10  
570-10  
570:10
```

You can also use this syntax to navigate to the Shared Components page of a given application:

```
application_ID:s  
application_ID-s
```

Where:

- `application_ID` is the numeric identification number (or ID) that identifies an application.

The following example demonstrates how to navigate to the Shared Components page of application 570:

```
570:s  
570-s
```

About Navigating Search Results

Global Search results display in a dialog. Use the Up and Down arrows keys to navigate the results and press **Enter** to make a selection. Since the the **Global Search** icon displays on nearly every page in Application Express, you can continue to narrow your search until you find the app, page, or component (such as the item, button, or region) which contains the keyword.

Other available shortcuts change depending on where you are in App Builder, for example:

- In Page Designer - To search for a page, press **Ctrl + 4**.
- In App - To search for as app, press **Ctrl + 3**.
- All Apps- To search all applications, press **Ctrl + 2**.
- Current Workspace- To search a workspace, press **Ctrl + 1**.

Example 1-1 Search Example

Suppose you need to find an application called *Employees* which has a report page built on the EMP table.

1. Find the application.

- a. On the Workspace home page, enter EMP in the Global Search field.

Global Search would return the results:

- Employees
- Search Workspace for EMP



- b. Select **Employees** to view that application.

The Employees application appears.

2. Search the application.

- a. On the Application home page, enter EMP in the Global Search field again.

Global Search returns the results that include numerous pages and other items.

- b. Select a page.

The page appears in Page Designer.

3. Search the page.

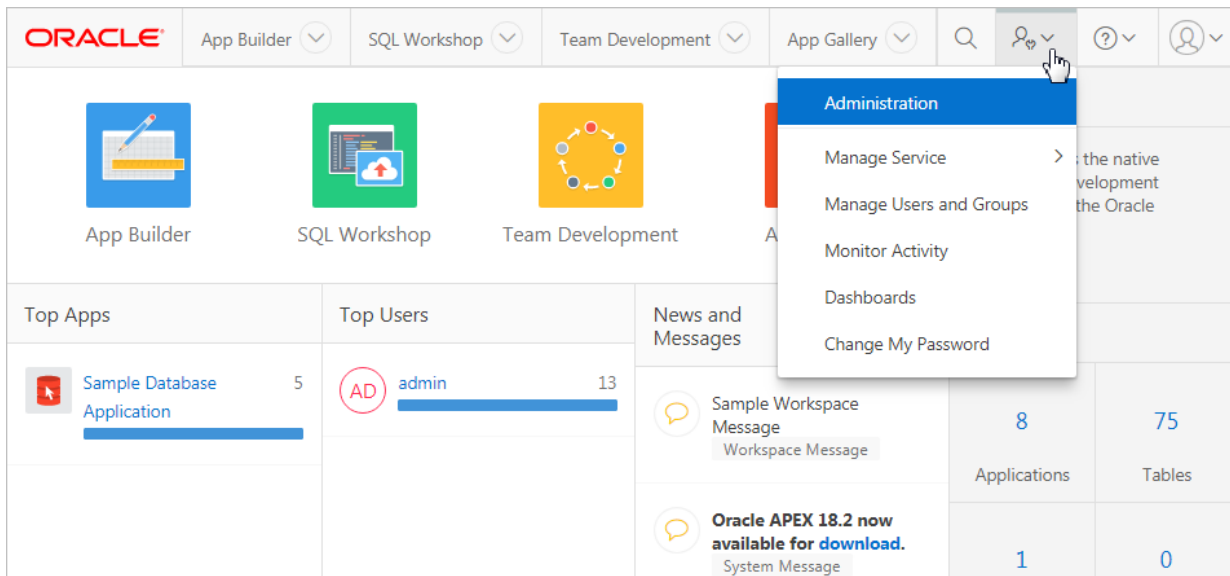
- a. In Page Designer, enter EMP in the Global Search field again.

Global Search returns the results that include numerous pages and other items.

- b. Select a component (such process, item, or region).

Administration Menu

The Administration menu displays to the right of the Global Search field and features an outline person with a wrench.



Clicking the Administration menu displays the following options:

- **Administration** links to the Workspace Administration page. To access the Oracle Application Express development environment, users sign in to a shared work area called a **workspace**. Workspace administrators monitor and administer a workspace. Common workspace administration tasks include managing service requests, configuring workspace preferences, creating workspace announcements, managing session state, creating user accounts, and managing groups.
- **Manage Service** displays a submenu of options on the Manage Service page. Administrators use the Manage Service page to manage service requests, configure workspace preferences, edit workspace announcements, and view workspace utilization reports.
- **Manage Users and Groups** links to Manage Users and Groups page. Workspace administrators can create new user accounts, manage existing user accounts, and change user passwords.
- **Monitory Activity** links to the Monitor Activity page. Use this page to monitor changes to page views, developer activity, page view analysis, and active sessions. It also shows reports on such information as login attempts and external click counts.
- **Dashboards** links to the Dashboard page. Use this page to view details about the current workspace and monitor users, activity, developer activity, performance,

Worksheet application development, application development, and view database details.

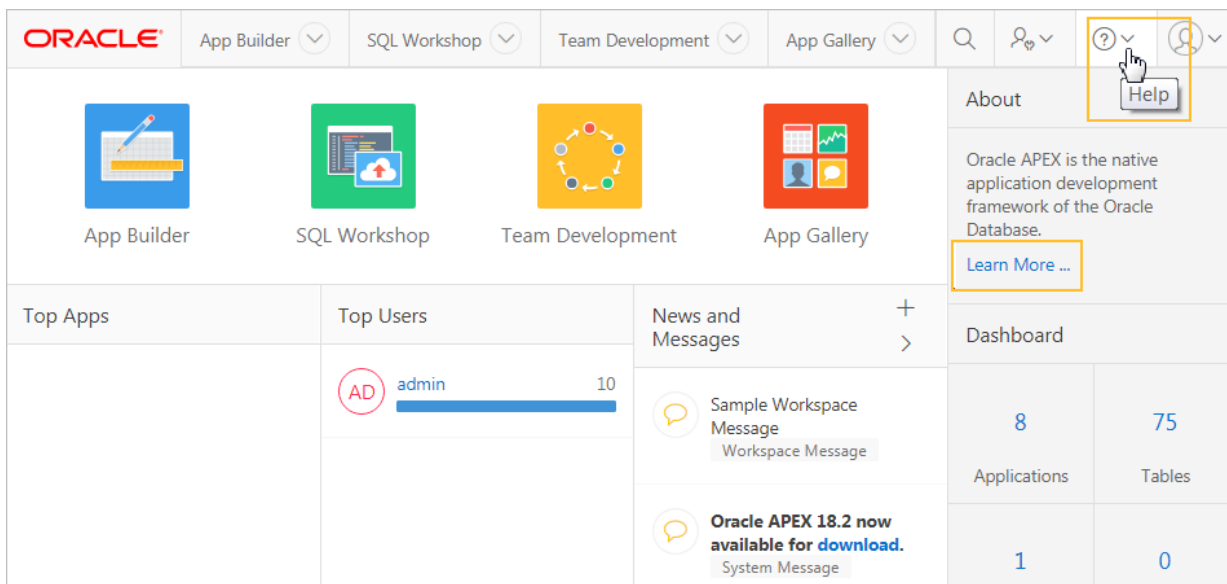
- **Change My Password** links to the Change Password section of the Edit Profile dialog box.

 **See Also:**

- "Workspace and Application Administration" in *Oracle Application Express Administration Guide*
- "Manage Service Page" in *Oracle Application Express Administration Guide*
- "Managing Users in a Workspace" in *Oracle Application Express Administration Guide*
- "Monitoring Activity Within a Workspace" in *Oracle Application Express Administration Guide*
- "Viewing Workspace Dashboards" in *Oracle Application Express Administration Guide*
- ["Changing Your Profile or Password"](#)

Help Menu

The **Help** menu features a question mark enclosed with a circle.



The screenshot shows the Oracle APEX workspace home page. At the top, there is a navigation bar with the Oracle logo and several application icons: App Builder, SQL Workshop, Team Development, and App Gallery. To the right of these icons is a search bar and a user profile icon. A yellow box highlights a question mark icon in a circle, which is the Help menu icon. Below the navigation bar, there is a main content area with several sections: 'About' (with a 'Learn More ...' link), 'Top Apps', 'Top Users' (showing a user named 'admin' with a score of 10), 'News and Messages' (with a list of messages), and 'Dashboard' (with a table showing 8 Applications and 75 Tables, and 1 message and 0 tables).

Click the Help menu to access the following menu items:

- **Documentation** links to Oracle Application Express, Oracle Help Center.

- **Discussion Forum** links to the [Application Express Discussion Forum](#).
- **Oracle Technology Network** links to the [Oracle Application Express page](#).
- **About**. Links links to the About Application Express dialog. This dialog includes information about the Oracle Application Express instance and the database version.



See Also:

["Using Oracle Application Express Documentation"](#)

Account Menu

The Account menu displays in the upper right corner of the page and features an outline of a person enclosed with a circle.

The screenshot shows the Oracle Application Express Workspace Home Page. At the top, there is a navigation bar with the Oracle logo and several application tiles: App Builder, SQL Workshop, Team Development, and App Gallery. Below the navigation bar, there are three large icons for App Builder, SQL Workshop, and Team Development. On the right side, the Account Menu is displayed, featuring a profile picture placeholder, the username 'ADMIN', email 'admin@example.com', workspace 'DOC_TEST', and role 'Administrator'. Below the profile information, there are buttons for 'Edit Profile', 'Preferences', and 'Sign out'. The main content area is divided into sections: 'Top Apps', 'Top Users', and 'News and Messages'. The 'Top Users' section shows a bar chart for the user 'admin' with a count of 11. The 'News and Messages' section displays a 'Sample workspace message' and a system message about 'Oracle APEX 18.2 now available for download'. At the bottom right, there is a summary table for 'Applications' and 'Tables'.

| Applications | Tables |
|-------------------|----------|
| 1 | 34 |
| Productivity Apps | Features |
| 0 | 0 |

Click the Account menu to view your username, current workspace, and user role. If Single Sign On (SSO) is enabled, the link **Change Workspace** also appears. Key controls on this menu include:

- **Edit Profile**. Click **Edit Profile** to edit your first and last name, upload or choose a photo image, alter your email address (if SSO is not enabled), and reset your password.

 **Tip:**

If you are using Single Sign-on authentication, **Change Workspace** also displays. Click **Change Workspace** to view a list of workspaces you have access to.

- **Preferences.** Click **Preferences** to change your default schema, default date format, and control how App Builder runs applications.
- **Sign Out** - Click **Sign Out** to log out the current workspace.

 **See Also:**

- ["Changing Your Profile or Password"](#)
- ["Editing User Preferences"](#)

Available Updates Region

The Available Updates region indicates if updates are available for Oracle Application Express and the Oracle REST Data Services. If your installation is current the following message displays:

System is up-to-date

To configure updates, click **Set Preferences for Available Updates**. You can enable or disable updates by selecting Yes or No from Check for Available Updates.

 **Tip:**

To perform this check Application Express transmits the version number of Oracle Application Express and other system components to Oracle Corporation. In order to continuously improve our products, Oracle is interested in learning about product usage. To that end, statistics can occasionally be sent to Oracle describing the product features in use. No personally identifiable information is sent. To review Oracle's privacy policy see <http://www.oracle.com/us/legal/privacy/overview/index.html>

Understanding App Builder

Developers use App Builder to create and manage applications and application pages.

The App Builder home page displays all installed applications in the current Oracle Application Express instance. When a developer selects an application to edit, the Application home page appears. Use the Application home page to run, edit, import, export, copy, or delete applications.

- [Accessing App Builder](#)
Access App Builder by clicking the App Builder icon on the Workspace home page.
- [Understanding the App Builder Home Page](#)
Use App Builder to create new applications, select an application to edit, import previously exported applications, view the Dashboard, or access workspace utilities.
- [Accessing the Application Home Page](#)
Access the Application home page by clicking the App Builder icon on the Workspace home page and then selecting an application.
- [Understanding the Application Home Page](#)
Use the Application home page to run, edit, import, export, copy, or delete applications.



See Also:

"Quick Start" and "App Builder Concepts"

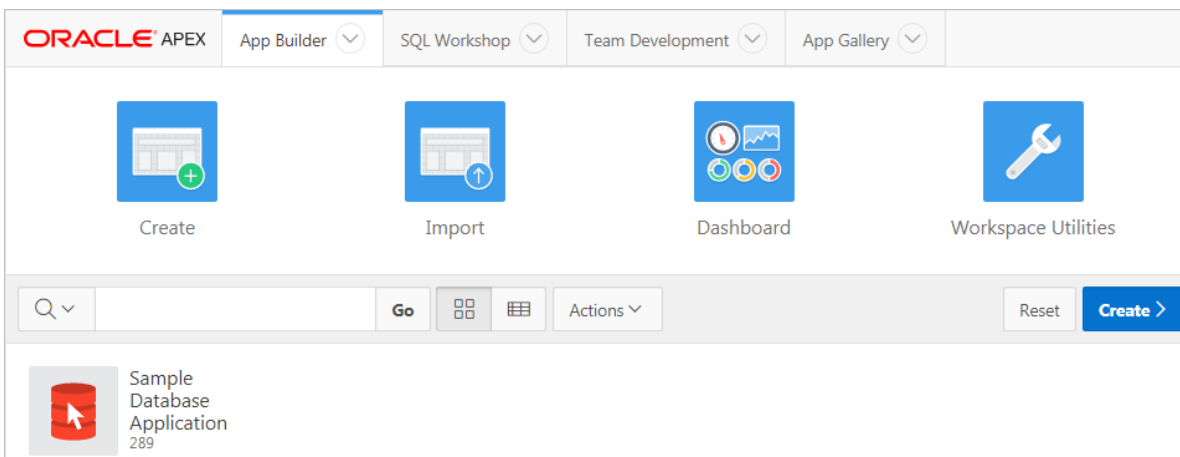
Accessing App Builder

Access App Builder by clicking the App Builder icon on the Workspace home page.

To access App Builder:

1. Sign in to Oracle Application Express.
2. On the Workspace home page, click the **App Builder** icon.

The App Builder home page appears.



The App Builder home page displays currently installed applications as shown in the previous illustration.

 **Tip:**

In a new workspace, the bottom of the page displays two additional regions. Click **Create a New App** to access the Create Application Wizard and create a new application. Click **Install a Productivity or Sample App** to access the App Gallery page and install a productivity or sample app and load sample data. Once you create an application or install a productivity or sample application, these regions no longer appear.

 **See Also:**

- ["Understanding the App Builder Home Page"](#)
- ["Creating Database Applications"](#)
- ["Installing a Productivity and Sample App"](#)

Understanding the App Builder Home Page

Use App Builder to create new applications, select an application to edit, import previously exported applications, view the Dashboard, or access workspace utilities.

 **Tip:**

In a new workspace, the bottom of the page displays two additional regions. Click **Create a New App** to access the Create Application Wizard and create a new application. Click **Install a Productivity and Sample Apps** to access the App Gallery page and install a Productivity and Sample App and load sample data. Once you create an application or install a sample or productivity application, these regions no longer appear.

- [App Builder Home Page Icons and Navigation Bar](#)
The top of the App Builder home page features four large icons (Create, Import, Dashboard, and Workspace Utilities) and the bottom of the page displays all currently installed applications.
- [Reset and Create Buttons](#)
The Reset and Create buttons display in the center of the page to the right of the navigation bar.
- [Recent List, Tasks List, and Migrations Link](#)
The App Builder home page feature three regions on the right side of the page: Recent, Tasks, and Migrations.
- [Workspace Utilities](#)
Use the Workspace Utilities page to manage App Builder Defaults, REST Enabled SQL services, news items, Remote Server objects, application groups, Web

credentials, workspace themes, application exports, Application Express views, and cross application reports.

- [Viewing Cross Application Reports](#)
Use the links under Cross Application Reports page to view and edit information across multiple applications in the current workspace.



See Also:

["Accessing App Builder"](#) and ["Developer Navigation Tools"](#)

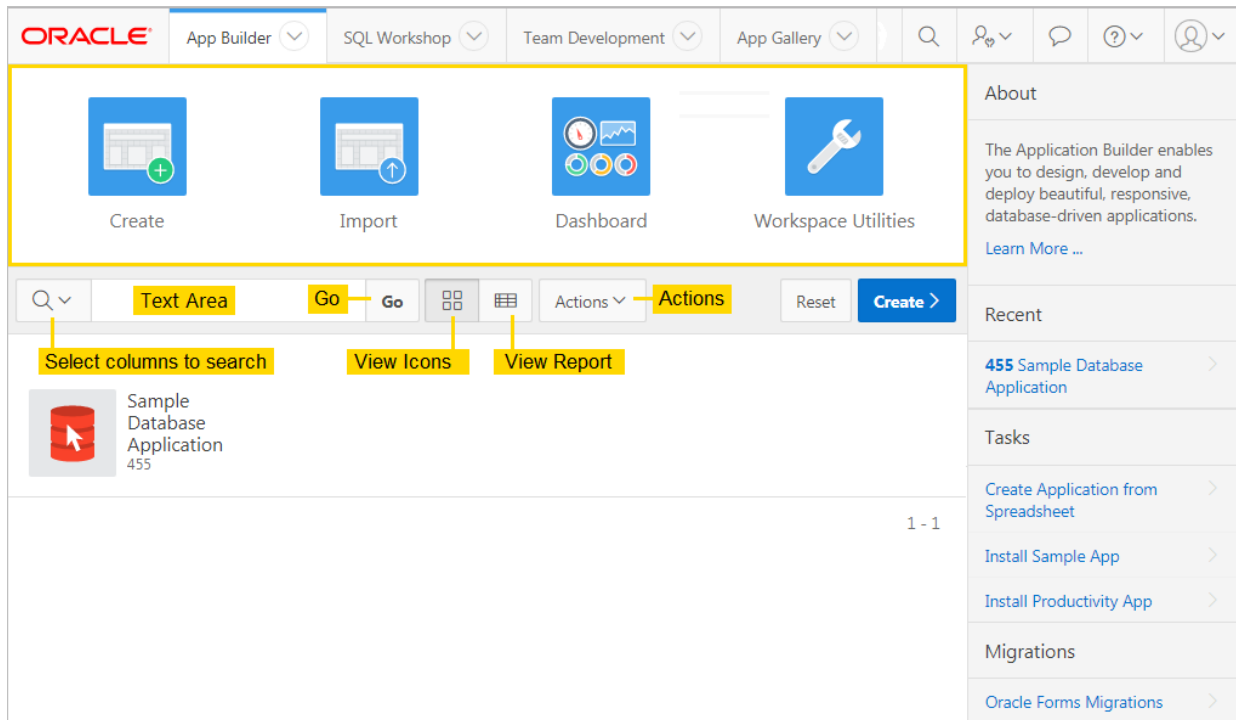
App Builder Home Page Icons and Navigation Bar

The top of the App Builder home page features four large icons (Create, Import, Dashboard, and Workspace Utilities) and the bottom of the page displays all currently installed applications.

About App Builder Home Page Icons

The top of App Builder home page contains four large icons:

- **Create.** Launches the Create Application Wizard. Use the Create Application Wizard to build a complete application containing multiple pages.
- **Import.** Launches the Import Wizard. Use this wizard to import export files from the export repository.
- **Dashboard.** Links to the App Builder Dashboard which displays metrics about applications in the current workspace.
- **Workspace Utilities.** Use the Workspace Utilities page to manage App Builder Defaults, news items, workspace themes, application groups, Application Express views, application exports, credentials, REST Enabled SQL services, Remote Server objects, and cross application reports.



About the Navigation Bar

The bottom of App Builder home page displays all installed applications. You can customize how the applications display using the navigation bar in the center of page.

Available controls include:

- **Select columns to search** - This field contains a magnifying glass icon. Click this icon to narrow your search to only specific columns. To search all columns, select **All Columns**.
- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search or applies a filter.
- **View Icons** (default) - Displays each application as a large icon identified by the application name.
- **View Report** - Displays each application as a line in a report. Each line includes the application ID, the application name, when the application was last updated, the page count, and who last updated the application.
- **Actions menu** - Displays the Actions menu. Use this menu to customize the report view.

 **Tip:**

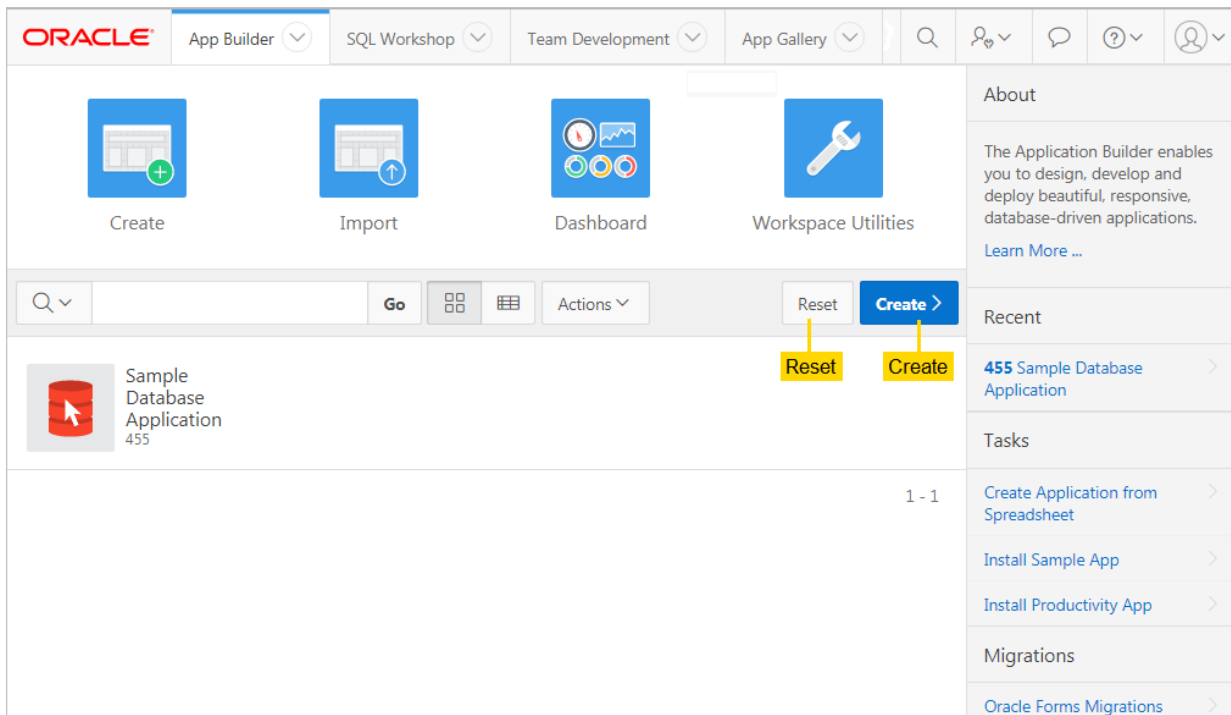
In a new workspace, the bottom of the App Builder home page displays two regions: **Create a New App** and **Install a Productivity or Sample App**. Click **Install a Productivity or Sample App** to access the App Gallery page and install a Productivity and Sample App and load sample data. utilities. Once you create an application or install a packaged application, these regions disappear.

 **See Also:**

- "About the Create Application Wizard"
- "Importing Export Files"
- "About the Actions Menu"

Reset and Create Buttons

The Reset and Create buttons display in the center of the page to the right of the navigation bar.

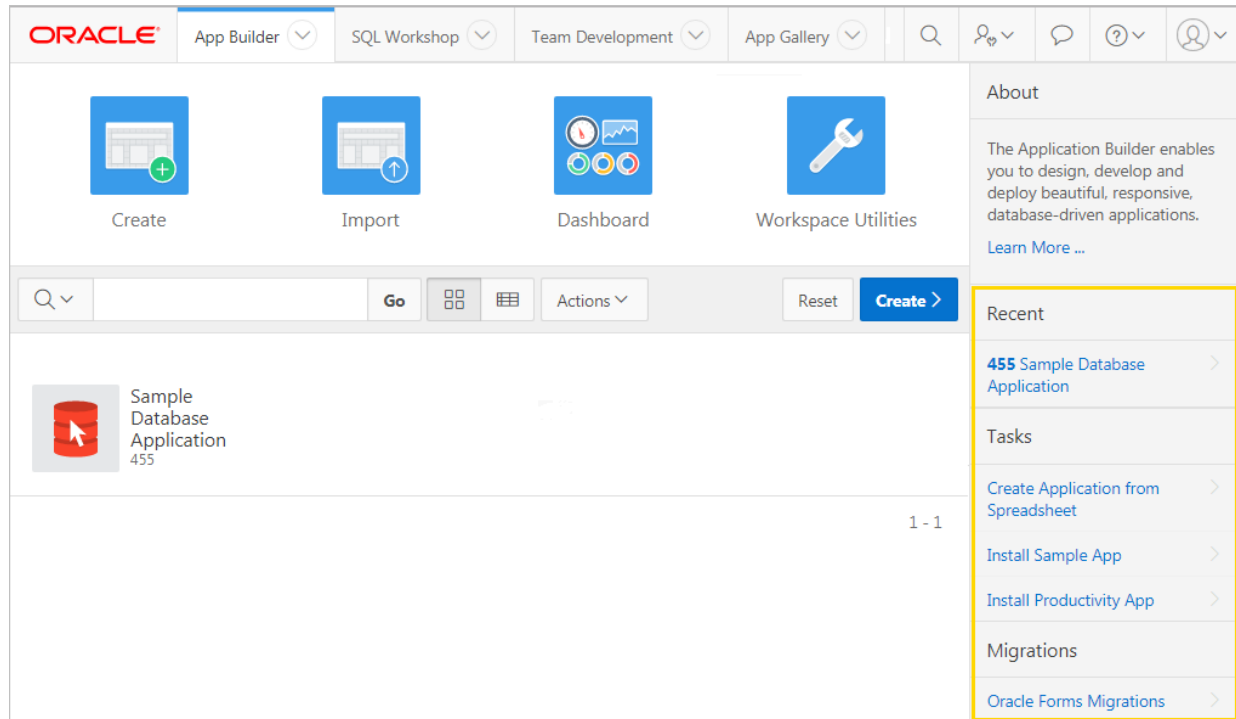


Use the **Reset** and **Create** buttons as follows:

- Click **Reset** to return the page to the default display.
- Click **Create** to create a new application, create an application based on a spreadsheet, install a productivity app, create a Worksheet app, launch Quick SQL, or copy an application.

Recent List, Tasks List, and Migrations Link

The App Builder home page feature three regions on the right side of the page: Recent, Tasks, and Migrations.



Recent

The **Recent** region contains links to recently viewed applications.

Tasks

The **Tasks** region displays on the middle of the right side of the page and features the following links:

- **Create Application From Spreadsheet** - Load spreadsheet data from a spreadsheet by importing it from a file as comma separated (*.csv) or tab delimited file, or by copying and pasting tab delimited data. Once you approve a page preview, the wizard loads the data into the database, creates a report and form on the data, and then displays the Create Application Wizard to complete the application creation process.
- **Install Sample App** - Click the link and select a sample application to install in your workspace.
- **Install Productivity Links** - Click the link and select a productivity application to install in your workspace.

 **See Also:**

- ["Creating a Database Application from a Spreadsheet"](#)
- ["About App Gallery"](#)
- ["Installing a Productivity and Sample App"](#)

Migrations

The **Migrations** region displays on the bottom of the right side of the page. Click the Migrations link to access the Oracle Application Express Application Migration Workshop. Use Migration Workshop to migrate a Microsoft Access application or convert an Oracle Forms application to an Oracle Application Express application.

 **See Also:**

Oracle Application Express Application Migration Guide

Workspace Utilities

Use the Workspace Utilities page to manage App Builder Defaults, REST Enabled SQL services, news items, Remote Server objects, application groups, Web credentials, workspace themes, application exports, Application Express views, and cross application reports.

To access Workspace Utilities, click the **Workspace Utilities** icon in the center of the App Builder home page.

The following table describes the links on the Workspace Utilities page.

Table 1-1 Links on the Workspace Utilities Page

| Link | Description | To Learn More |
|---------------------------|--|---|
| App Builder Defaults | Define application settings, themes, and globalization attributes. | See Leveraging App Builder Defaults in <i>Oracle Application Express Administration Guide</i> |
| REST Enabled SQL Services | Manage references to external REST Enabled SQL services. | See: " Managing REST Enabled SQL References " |
| Manage News | Manage news displayed on the Application Express home page and Team Development. | See: " Managing News Entries " |
| Remote Servers | Manage Remote Server objects used in Web Source Modules and Authentications. | See: " Managing Remote Servers " |

Table 1-1 (Cont.) Links on the Workspace Utilities Page

| Link | Description | To Learn More |
|---------------------------|--|--|
| Application Groups | Organize applications by assigning them to application groups. To use application groups, first create a group and then assign applications to it. | See: " Managing Application Groups " |
| Web Credentials | Manage secure credentials to connect to REST Enabled SQL or other REST services. | See: " Managing Web Credentials " |
| Workspace Themes | Manage workspace themes that can be utilized by any application within the workspace. | See: " Managing Workspace Themes " |
| Application Express Views | Query the various views against Application Express meta data. | See: " Accessing Application Express Views " |
| Export | Export workspace components. | See: " Exporting an Application and Application Components " |
| Cross Application Reports | View and edit information across multiple applications in the current workspace. | See: " Viewing Cross Application Reports " |

Viewing Cross Application Reports

Use the links under Cross Application Reports page to view and edit information across multiple applications in the current workspace.

To access cross application reports:

1. Sign in to Oracle Application Express.
The Workspace home page appears.
2. Click the **App Builder** icon.
The App Builder home page appears.
3. Click the **Workspace Utilities** icon.
4. Locate the **Cross Application Reports** region on the right side of the window.
Available reports include:
 - All Reports
 - Application Attributes
 - Build Options
 - Developer Comments
 - Comments Calendar
 - Application Comparison
 - Component Comparison
 - Security Profiles

- Authentication Schemes
 - Password Items
 - Theme Summary
5. To view all reports:
 - a. Click **All Reports**.
 - b. On the Cross Application Reports page, select a report to review.

Accessing the Application Home Page

Access the Application home page by clicking the App Builder icon on the Workspace home page and then selecting an application.

Use the Application home page to run, edit, import, export, copy, or delete applications.

To access the Application home page:

1. Sign in to Oracle Application Express.
The Workspace home page appears.
2. Click the **App Builder** icon.
The App Builder home page appears.
3. Select an application on the Application home page.
The Application home page appears.



See Also:

["Understanding the Application Home Page"](#)

Understanding the Application Home Page

Use the Application home page to run, edit, import, export, copy, or delete applications.

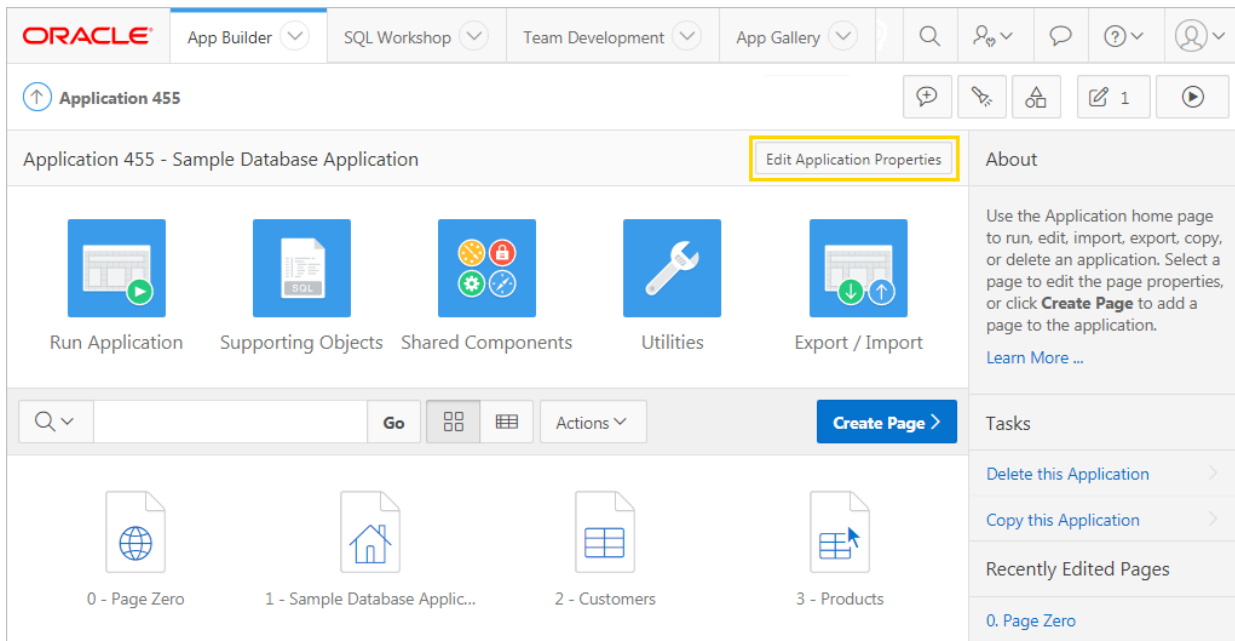
When a developer selects an application on the App Builder home page, the Application home page appears.

- [Edit Application Properties Button](#)
- [Application Home Page Icons](#)
- [Create Page Button](#)
- [Application Home Page Navigation Bar](#)
- [Application Home Page Tasks List](#)
- [Recently Edited Pages](#)

 **See Also:**

"Accessing App Builder" and "Developer Navigation Tools"

Edit Application Properties Button



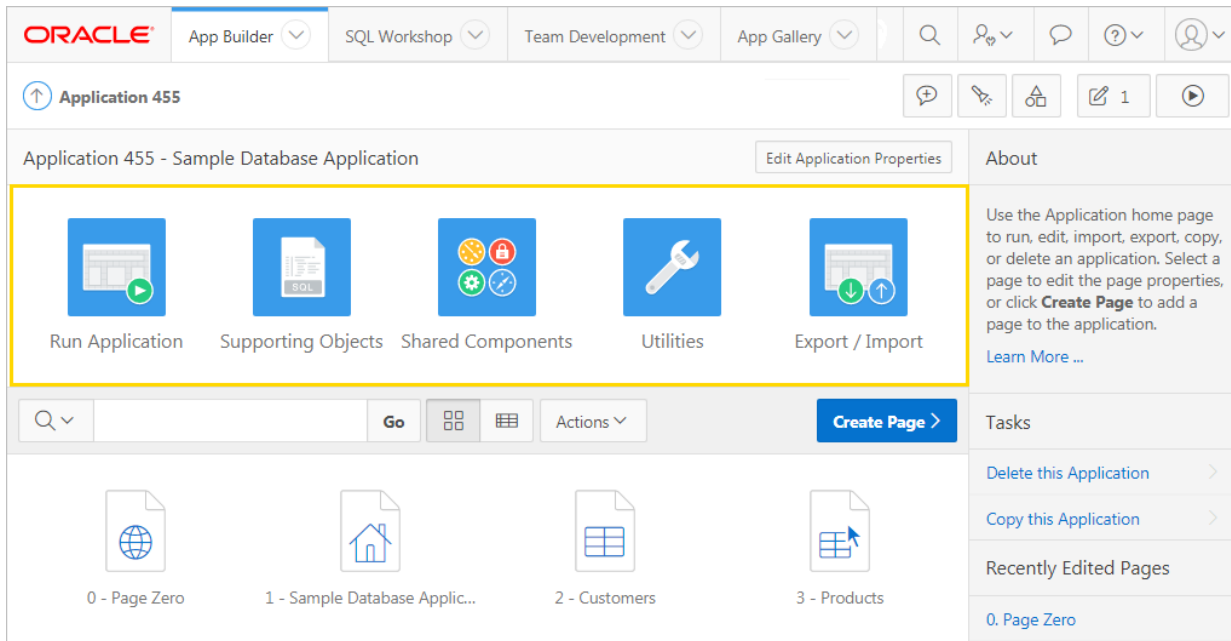
The application ID and the application name display at the top of the Application home page. The application ID also displays in the breadcrumb at the top of the page.

The Edit Application Properties button displays to the right of the application ID and name. Click **Edit Application Properties** to edit the application definition, including the application Name and Application Alias.

 **See Also:**

"Editing Application Attributes"

Application Home Page Icons



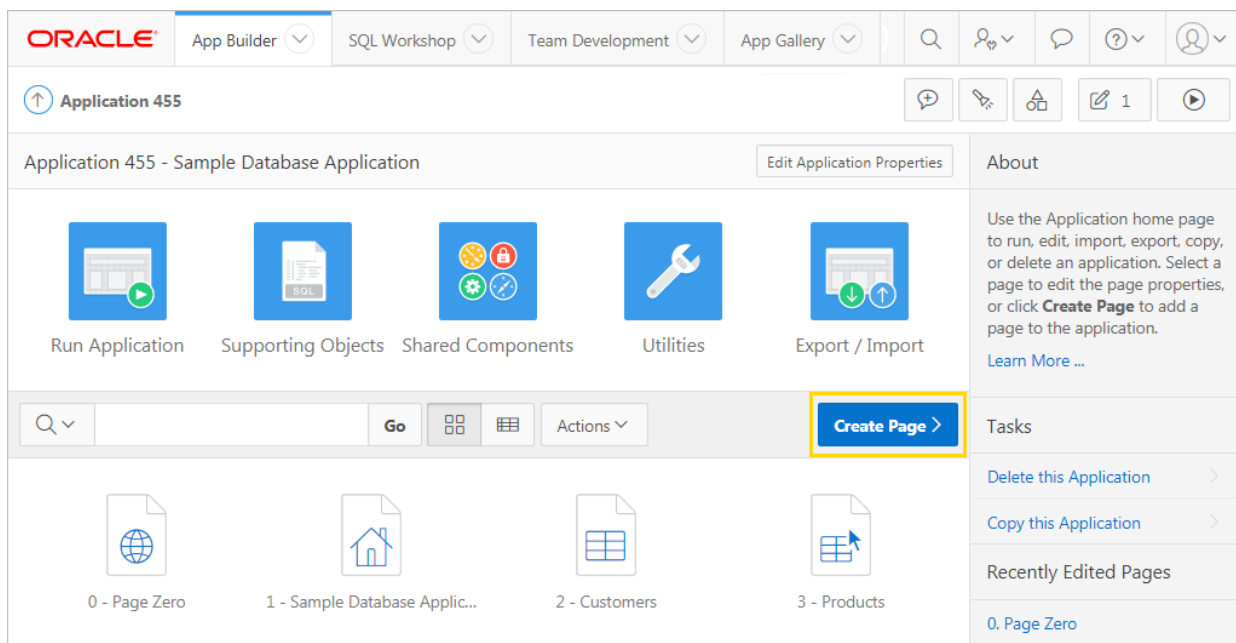
The following large icons appear beneath the application ID and application name:

- **Run Application** submits the home page in the current application to the Application Express engine to render viewable HTML.
- **Supporting Objects** links to the Supporting Objects page. Use Supporting Objects to define database object installation scripts that are invoked when importing an application. You can also define deinstallation scripts to drop objects when deleting an application.
- **Shared Components** links to the Shared Components page. Shared components can display or be applied on any page within the application.
- **Utilities** links to the Utilities page. Use this page to monitor developer activity, view dashboards, run Advisor, and view numerous other reports.
- **Export/Import** links you to the Export/Import Wizard. Use this wizard to import and export an entire application and related files such as cascading style sheets, images, static files, script files, themes, user interface defaults, and workspace users.

 **See Also:**

- "Running an Application or Page"
- "How to Create a Custom Application"
- "Managing Shared Components"
- "Using Application Utilities"
- "Exporting an Application and Application Components"

Create Page Button



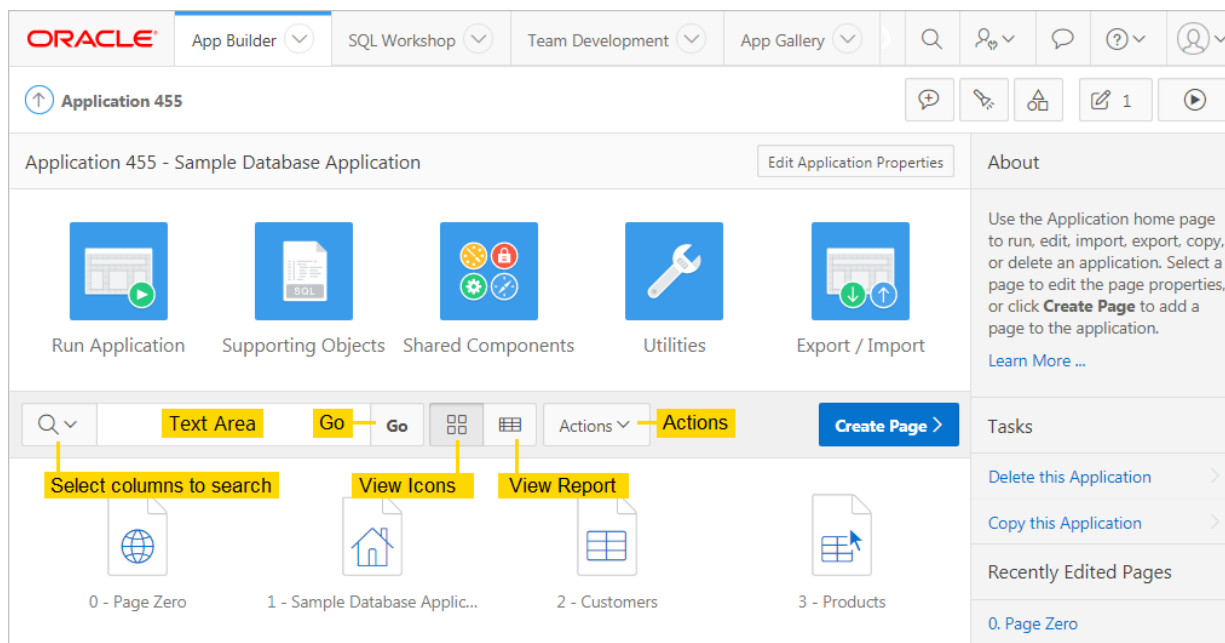
Clicking the **Create Page** button launches a wizard that walks you through creating a page in an application.

 **See Also:**

- "Adding a New Page to an Application"

Application Home Page Navigation Bar

A navigation bar displays in the center of the Application home page. You can use the navigation to search for pages or customize the appearance of the page.



The Application home page navigation bar contains the following controls:

- **Select columns to search** - This field contains a magnifying glass icon. Click this icon to narrow your search to only specific columns. To search all columns, select **All Columns**.
- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search or applies a filter.
- **View icons**. Use these icons to toggle between three views:
 - **View Icons** (default) - Displays each page as a large icon identified by the page name.
To view a page, click the page icon.
 - **View Report** - Displays each page as a line in a report. Each line includes the page number (ID), the page name, when the page was last updated and by whom, the page type, any associated group, the user interface, and lock status.
To view a page, click the page name. Click the **Lock** icon to prevent conflicts during application development. Click the **Run** icon to run the associated page and render viewable HTML.
- **Actions menu** - Displays the Actions menu. Use this menu to customize the report view.

 **See Also:**

- ["App Builder Home Page Icons and Navigation Bar"](#)
- ["Locking and Unlocking a Page"](#)

Application Home Page Tasks List

A Tasks list displays on the right side of the Application home page.

The Tasks list contains the following links:

- **Delete this Application** deletes the current application.
- **Copy this Application** creates a copy of the current application.

 **See Also:**

- ["Deleting an Application"](#)
- ["Copying a Database Application"](#)

Recently Edited Pages

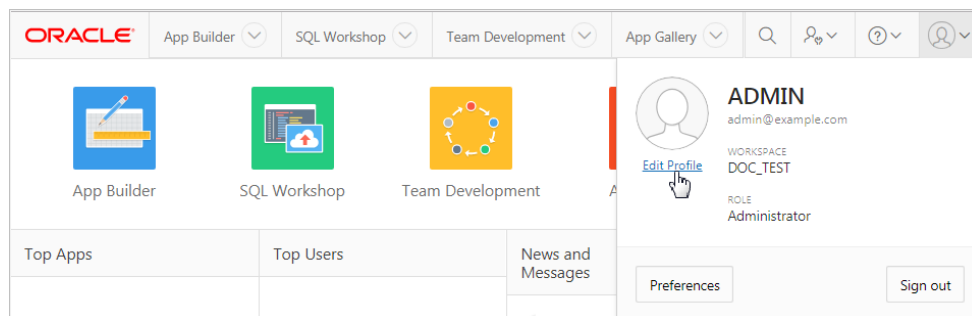
Recently Edited Pages region contains links to recently edited pages within the current application. To link to specific page, click the page ID and name.

Changing Your Profile or Password

Edit your account profile to update your email address, edit your first or last name, upload a profile photo, or change your password.

To edit your account profile:

1. Click the **Account** menu in the header region.
2. Click **Edit Profile**.



The Edit Profile dialog appears.

- Under Profile Details, edit the following attributes:
 - Email Address
 - First Name
 - Last Name

To learn more about an attribute, see field-level Help

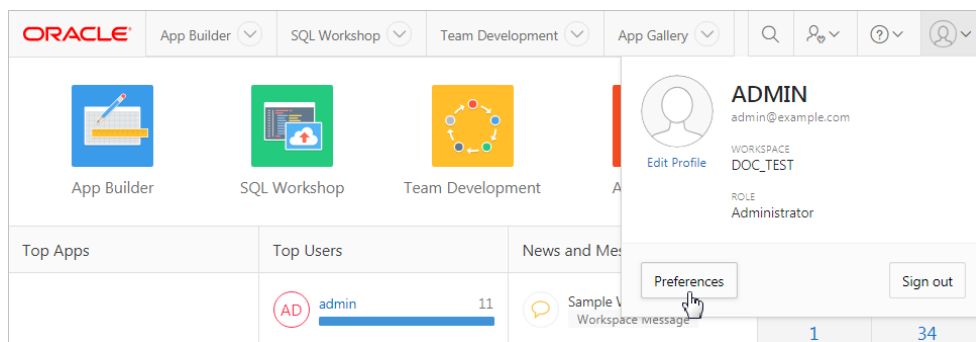
- Under Profile Photo, add, change, or remove your photo.
- Under Change Password, enter a new password.
- Click **Apply Changes**.

Editing User Preferences

Edit your user preferences to update the default workspace schema, specify a default date format, and control how Oracle Application Express manages windows and tabs when running applications.

To edit your user preferences:

- Click the **Account** menu in the header region and click **Preferences**.



The Preferences dialog appears.

- Edit the following attributes:
 - Default Schema** - Specifies the default schema used for data browsing, application creation, and SQL script execution. When using workspaces that have more than one schema available, this schema is the default for creating applications, performing SQL queries, and so on. This feature does not control security, only the user's preference.
 - Default Date Format** - Specify the default Oracle date format for the user. This will control the default date format within SQL Workshop.
 - Run application in new window** - Specify how Oracle Application Express manages windows or tabs when running applications from App Builder.

This option only impacts you when using Mozilla Firefox and Microsoft Internet Explorer, and has no effect when using other browsers. Options:

- Yes** - The default, browser preferences are ignored and a new window always opens.

- **No** - The runtime application opens in a new window or tab based on Browser preferences. Mozilla Firefox and Microsoft Internet Explorer generally do not set focus to another tab in the same window, except when that tab is first launched.
 - Use single window to run all applications - Specify whether different applications reuse the same application runtime window or tab. Options:
 - **Yes** - Each application you run from App Builder reuses a window or tab used for running the previous application.
 - **No** - Each application you run is opened in its own window or tab.
3. Click **Apply Changes**.

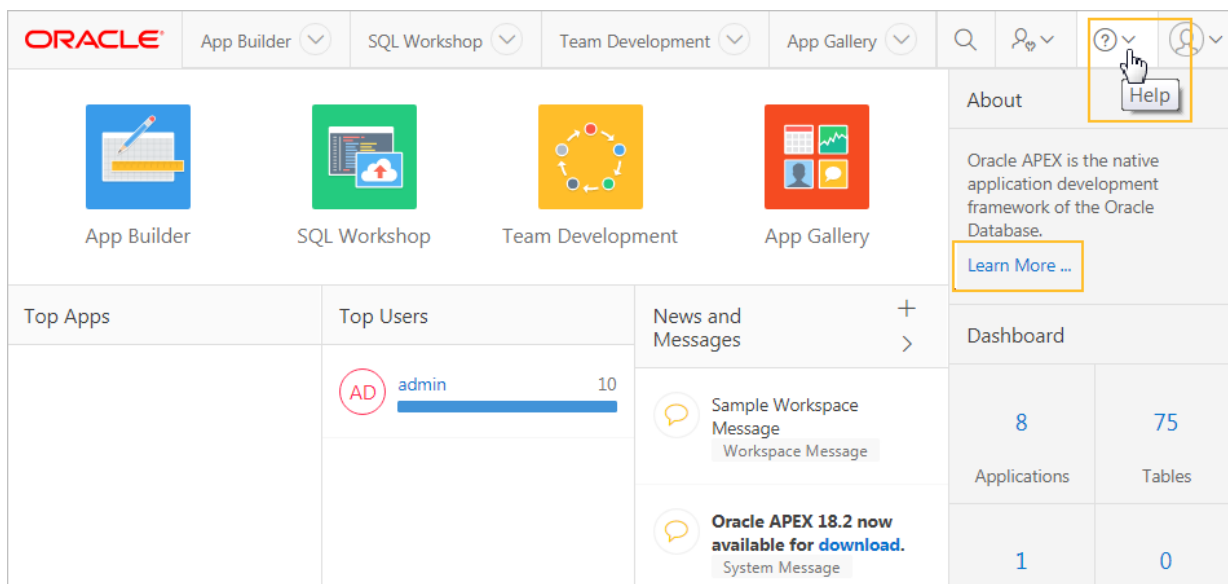
Using Oracle Application Express Documentation

Oracle Application Express provides a hosted online documentation library and field-level Help. You access, search, or download the online documentation library by going to the Oracle Help Center.

- [Accessing Oracle Help Center](#)
- [Searching the Oracle Help Center](#)
- [Downloading the Documentation Library](#)
- [Viewing Field-Level Help](#)

Accessing Oracle Help Center

Oracle Help Center offers a comprehensive library of publications that describe how to use Oracle Application Express. Whether you are new to Oracle Application Express or an advanced user, you can find useful information about Oracle Application Express, ranging from getting started guides to advanced features.



The screenshot shows the Oracle APEX dashboard interface. At the top, there is a navigation bar with the Oracle logo and several application-specific dropdown menus: App Builder, SQL Workshop, Team Development, and App Gallery. To the right of these menus are search and user profile icons. A yellow box highlights the 'Help' icon, which is a question mark inside a circle. Below the navigation bar, there are four main application tiles: App Builder, SQL Workshop, Team Development, and App Gallery. To the right of these tiles is an 'About' section for Oracle APEX, which includes a 'Learn More ...' link. Below the 'About' section is a 'Dashboard' section with a table showing the number of Applications and Tables. The table has two columns: Applications and Tables. The first row shows 8 Applications and 75 Tables. The second row shows 1 Application and 0 Tables.

| Applications | Tables |
|--------------|--------|
| 8 | 75 |
| 1 | 0 |

To access Oracle Help Center:

- Click the **Help** menu in the upper right corner and select **Documentation**.
Oracle Help Center appears.
- Click **Learn More...** links within the user interface to view information regarding the current page.
Clicking a **Learn More...** link displays information regarding the current page.

Redirecting the Help Menu

Administrators can redirect the Help menu at the instance-level in **Manage Instance, Instances Settings** in the Oracle Application Express Administration Services application. Oracle Application Express Administration Services is a separate application that enables an Instance administrator to manage an entire Oracle Application Express hosted instance. This functionality is not available to some customers who are using a hosted instance.



See Also:

“Configuring the Help Menu” in *Oracle Application Express Administration Guide*

Searching the Oracle Help Center

To perform a keyword search of the online documentation library:

1. Access Oracle Help Center:
 - a. Click the **Help** menu in the upper right corner. The **Help** menu features a question mark enclosed with a circle.
 - b. Select **Documentation**.
Oracle Help Center appears.
2. To search the entire library:
 - a. Enter keywords in the Search field at the top of the page and press **Enter** (or click the **Search** icon).
Search results display.
 - b. Select a topic to view.
3. To search a specific document:
 - a. Enter keywords in the Search field in the left pane and press **Enter** (or click the **Search** icon).
Search results display.
 - b. Select a topic to view.

 **Tip:**

While viewing a topic, a table of content displays in the left pane. To hide this pane, click the left facing arrow beneath the document title. The left pane is temporarily hidden.

Downloading the Documentation Library

To download the online documentation library:

1. Access Oracle Help Center:
 - a. Click the **Help** menu in the upper right corner. The **Help** menu features a question mark enclosed with a circle.
 - b. Select **Documentation**.Oracle Help Center appears.
2. Under Download the Oracle Application Express Bookshelf, click **Download all Oracle Application Express release 18.2 books**.
3. Save the ZIP file locally.
4. Download the ZIP file and extract it locally.
5. Open the extracted folder and click **index.html**.

The Oracle Application Express Documentation Library welcome page appears.

Viewing Field-Level Help

Most select lists, check boxes, items, and fields within the Oracle Application Express include field-level Help. Field-level Help displays in two different ways depending upon where you are viewing it.

- [Viewing Field-level Help in Oracle Application Express](#)
- [Viewing Help in Page Designer](#)

Viewing Field-level Help in Oracle Application Express

Most select lists, check boxes, items, and fields within the Oracle Application Express include field-level Help.

To view field-level help:

1. Navigate to the attribute you need information about.

Most attributes in Oracle Application Express include field-level Help. Attributes with field-level Help, have light gray icon that resembles a question mark (?). Passing your cursor over the icon displays the tooltip, **Help Text:**

2. To view field-level Help, click the **Help Text** icon.
A Help window appears.

Debugging

Determines whether debug mode can be enabled from a browser. Available options include:

- **Yes.** Enables the application to run in a debug mode from a browser at runtime.
- **No.** Disables the application from enabling in debug from a browser.

Running an application in debug mode is useful when an application is under development. For production applications, Oracle recommends disabling debugging and thus preventing users from viewing application logic..

Debug can be enabled programmatically regardless of this debug setting.

If the application is run from the Application Express development environment, debugging can always be enabled.

Tip:

To access field-level help with the keyboard, focus on the correct field and then press **Alt+F1**.

See Also:

["Accessing Oracle Help Center"](#)

Viewing Help in Page Designer

Page Designer includes Help for all properties in the Property Editor.

To view Help in Page Designer:

1. Access Page Designer.

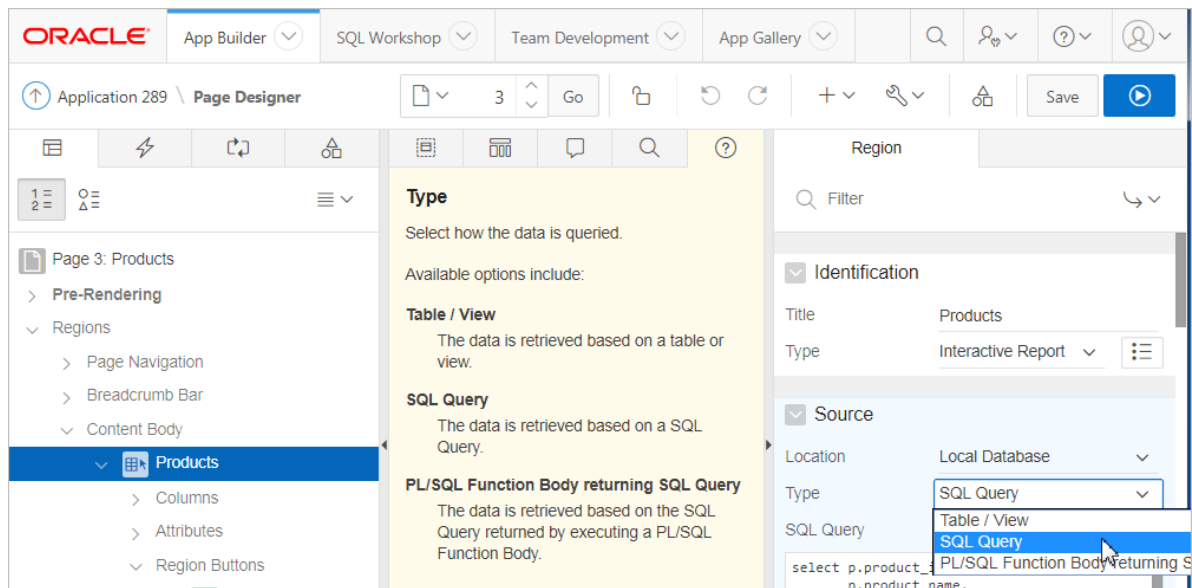
Page Designer appears.

2. Select a component or control in either the left pane or the central pane.

As you change focus in either view, the Property Editor automatically updates to reflect the currently selected component.

3. In the Property Editor, select an attribute.

4. To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.



Once you activate the Help pane, the content that displays changes every time you select another attribute.

Tip:

To view an online summary that describes just the Page Designer window, view a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.

See Also:

"Viewing a Page in Page Designer"

2

App Builder Concepts

To efficiently use App Builder, developers must understand some key concepts including managing user interface design, viewing rendered application page, understanding page processing and rendering, managing session state, understanding URL syntax, managing session state values, and using built-in substitution strings.

- [Understanding Oracle Application Express Applications](#)
Learn more about Oracle Application Express applications, the differences between database and Websheet applications, and the advantages of using the Create Application Wizard.
- [Running an Application or Page](#)
To view a rendered version of your application, you run or submit it to the Application Express engine. The Application Express engine dynamically renders and processes pages based on data stored in database tables.
- [How Does Page Processing and Page Rendering Work?](#)
Learn how the Application Express engine renders and processes pages, when Oracle Application Express issues implicit commits, how conditional rendering and processing works, and how to verify user identify or control access to controls and components.
- [About Enabling Support for Bookmarks](#)
Developers can assist users who bookmark application pages by either using zero as the session ID, or enabling Rejoin Sessions.
- [Understanding Session State Management](#)
Oracle Application Express transparently maintains session state and provides developers with the ability to get and set session state values from any page in the application.
- [Viewing Session State](#)
View the Session page by clicking **Session** on the Developer toolbar.
- [Understanding URL Syntax](#)
Use f?p syntax to link pages or call pages using a page alias or button URL.
- [Managing Session State Values](#)
Manage session state to store and retrieve values for a user as the user navigates between different application pages.
- [Using Substitution Strings](#)
Use substitution strings within a page template or region source to pass information and replace a character string with another value.



See Also:

" [Understanding App Builder](#) "

Understanding Oracle Application Express Applications

Learn more about Oracle Application Express applications, the differences between database and Websheet applications, and the advantages of using the Create Application Wizard.

- [What is an Oracle Application Express Application?](#)
- [What is the Difference Between a Database and Websheet Application?](#)



See Also:

["Understanding App Builder"](#)

What is an Oracle Application Express Application?

An Oracle Application Express application is an HTML interface that exists on top of database objects such as tables or procedures. You create both database applications and Websheet applications using the Create Application Wizard. The main difference between these two types of applications is the intended audience. While database applications are primarily developed by application developers, Websheet applications are often created by end users with no development experience.

What Is an Application Page?

A page is the basic building block of an application. Every application consists of one or multiple pages. Each page can have buttons and fields (called **items**) which are grouped into containers called **regions**. Pages can include application logic (or processes). You can branch from one page to the next using conditional navigation; perform calculations (called **computations**); perform validations (such as edit checks); and display reports, calendars, and charts. You view and edit a page in Page Designer.



See Also:

["Editing Pages in Page Designer"](#)

What is the Difference Between a Database and Websheet Application?

An Oracle Application Express application enables developers to manage and display data stored in an Oracle database. You build an application using App Builder. Using App Builder you can create two different types of applications: **Database applications** and **Websheet applications**.

About Database Applications

A database application is a collection of pages linked together using navigation menus, tabs, buttons, or hypertext links. Application pages share a common session state and authentication.

To create a database application, an application developer runs wizards to declaratively assemble pages and navigation. Individual pages are organized using containers called regions. Regions can contain text, custom PL/SQL, reports, charts, maps, calendars, web service content, or forms. Forms are made up of fields (called items) which can be selected from the multitude of built-in types (such as text fields, text areas, radio groups, select lists, check boxes, date pickers, and popup list of values).

Developers can also create their own custom item types using plug-ins. Session state (or application context) is transparently managed and the user interface presentation is separated from the application logic enabling developers to manage the look and feel of an application by simply selecting a different theme.

About Websheet Applications

Websheet applications enable end users to manage structured and unstructured data without developer assistance. Page sections contain unstructured data which can be edited using a WYSIWYG editor. Data Grids enable users to manage structured data without the need for writing SQL. Using runtime dialog boxes, users can add columns, rename columns, and validations. Each page and row of data grid data can be annotated with files, tags, notes, and links. Pages can contain sections, reports, and data grids and everything can be linked together using navigation. All information is searchable and completely controlled by the end-user.


See Also:

["Creating Database Applications"](#) and ["Creating Websheet Applications"](#)

Running an Application or Page

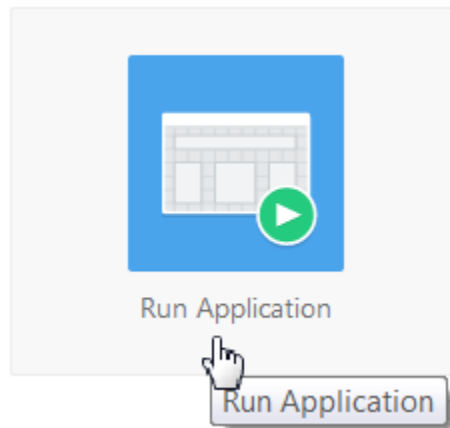
To view a rendered version of your application, you run or submit it to the Application Express engine. The Application Express engine dynamically renders and processes pages based on data stored in database tables.

- [Run Application and Run Page Buttons](#)
- [How Your Browser Impacts the Way Applications Run](#)
- [Running an Application from the App Builder Home Page](#)
- [Running an Application from the Application Home Page](#)
- [Running a Page on the Application Home Page](#)

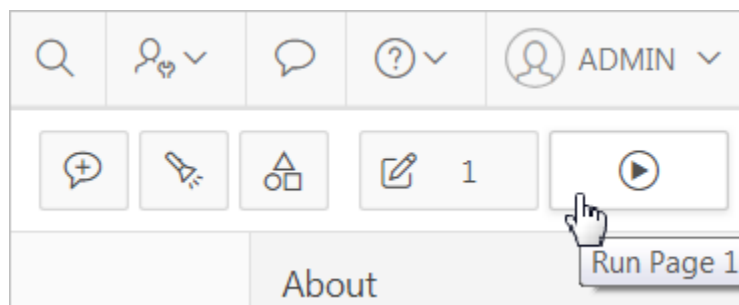
 **See Also:**
"Running a Page from Page Designer"

Run Application and Run Page Buttons

As you create new pages, you can run them individually, or run an entire application. The **Run Application** button displays on the Application home page and resembles an application window with a green run (or play) button. To run the application from the application home page, click **Run Application**.




The Run Page button resembles a small, gray play button and displays in the upper right corner of many pages within App Builder. Clicking the **Run Page** button runs the current page or the last selected page.



How Your Browser Impacts the Way Applications Run

When you run an application, the application displays in a new window. Whether that new window is a new tab or new browser windows depends upon how you have configured your web browser. If you are using Google Chrome or Apple Safari, the running application displays in a new tab by default. If you are using Microsoft Internet Explorer or Mozilla Firefox, the application displays in a new browser window by

default. You can further control how Oracle Application Express manages tabs and windows when running an application by editing your account preferences.

 **See Also:**
["Editing User Preferences"](#)

Running an Application from the App Builder Home Page

To run an application from the App Builder home page:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **View Report** icon (optional).

Selecting **View Report** displays each application as a line in a report. Each line includes the application ID, the application name, the number of application pages, and update information.

3. Locate the application to be run.
4. To run the application:
 - In Report view - Click **Run Page** icon in the far right column.
 - In Icon view - Click the **Run Application** icon at the top of the page.

The application displays in a new window.

 **See Also:**
["How Your Browser Impacts the Way Applications Run"](#)

Running an Application from the Application Home Page

To run an application from the Application home page:

1. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
2. Select an application.
3. Click the **Run Application** icon in the middle of page.
The application displays in a new window.

 **See Also:**
["How Your Browser Impacts the Way Applications Run"](#)

Running a Page on the Application Home Page

To run a page from the Pages list:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **View Report** icon.

Selecting **View Report** displays each page as a line in a report. Each line includes the page number, the page name, when the page was last updated, who last updated the page, the page type, the selected user interface, a lock icon, and a Run button.

4. Locate the page you want to run.
5. Click **Run Page** in the far right column.

The application displays in a new window.



See Also:

["Locking and Unlocking a Page"](#) and ["How Your Browser Impacts the Way Applications Run"](#)

How Does Page Processing and Page Rendering Work?

Learn how the Application Express engine renders and processes pages, when Oracle Application Express issues implicit commits, how conditional rendering and processing works, and how to verify user identity or control access to controls and components.

- [How the Application Express Engine Renders and Processes Pages](#)
- [About Implicit Commit Points](#)
- [Understanding Conditional Rendering and Processing](#)
- [About Verifying User Identity](#)
- [About Controlling Access to Controls and Components](#)

How the Application Express Engine Renders and Processes Pages

The Application Express engine dynamically renders and processes pages based on data stored in Oracle database tables. To view a rendered version of your application, you request it from the Application Express engine with a URL. When you run an application, the Application Express engine relies on two processes:

- **Show Page** - Show Page is a page rendering process that assembles all the page attributes (including regions, items, and buttons) into a viewable HTML page. When you request a page using a URL, the engine is running **Show Page**.
- **Accept Page** - Accept Page performs forms page processing, including computations, validations, processes, and branching. When you submit a page, the Application Express engine is running **Accept Page** or performing page

processing during which it saves the submitted values in the session cache and then performs any computations, validations, or processes.

About Implicit Commit Points

Oracle Application Express issues implicit commits at the following points:

- On load, after a page finishes rendering
- On submit, before branching to another page
- On submit, if one or more validations fail, before re-rendering the page
- After a computation
- After changing the value of an item, for example after a PL/SQL process that modifies a bind variable value or when `APEX_UTIL.SET_SESSION_STATE` is called.
- When `APEX_MAIL.PUSH_QUEUE` is called

See Also:

"`SET_SESSION_STATE` Procedure" and "`PUSH_QUEUE` Procedure" in *Oracle Application Express API Reference*

Understanding Conditional Rendering and Processing

Developers can use conditions in an application to control how pages and page components display and when processes, computations, and validations execute.

- [What is a Condition?](#)
- [Selecting a Condition](#)
- [Condition Example: Current Page in Expression 1](#)
- [Condition Example: Exists](#)
- [Condition Example: PL/SQL Expression](#)

What is a Condition?

A condition is a small unit of logic that helps you control the display of regions, items, buttons, and tabs and the execution of processes, computations, and validations. For example, when you apply a condition to a button, the rendering engine evaluates the condition during the rendering (or Show Page) process. Whether the condition passes or fails determines if the button displays.

See Also:

"[Available Conditions](#)" for a detailed listing of available condition types

Selecting a Condition

You specify a condition by selecting a condition type. You can select a condition type when you first create the control or component, or later when editing the control or component.

Depending upon the type of condition you select, you enter the appropriate values in the fields provided. The condition evaluates to true or false based on the values you enter. Click the select list to view a complete list of all available conditions for a given component or control.

Condition Example: Current Page in Expression 1

Current page in Expression 1 evaluates to true if the current page number is contained within the comma-delimited list of pages provided. For example:

```
3,100,203
```

If the current page is 100, then this condition evaluates to true and the condition passes.

Condition Example: Exists

Exists (SQL query returns at least one row) is expressed as a SQL query. If the query returns at least one row, then the condition evaluates as true. For example:

```
SELECT 1 FROM employees WHERE department_id = :P101_DEPTNO
```

This example references item P101_DEPTNO as a bind variable. You can use bind variables within application processes and SQL query regions to reference items from session state. If one or more employees are in the department identified by the value of P101_DEPTNO, then the condition evaluates as true.



See Also:

["Referencing Session State Using Bind Variable Syntax"](#)

Condition Example: PL/SQL Expression

Use **PL/SQL Expression** to specify an expression in valid PL/SQL syntax that evaluates to true or false. For example:

```
NVL(:MY_ITEM, 'NO') = 'YES'
```

If the value of :MY_ITEM is YES, as in the previous example, then the condition evaluates as true. If the value of :MY_ITEM is anything other than YES, then the condition evaluates as false.

About Verifying User Identity

Authentication is the process of establishing users' identities before they can access an application. Authentication may require a user to enter a user name and password, or may involve the use of a digital certificate or a secure key.

Oracle Application Express supports modular authentication, making it easy to switch authentication methods when needed. You can establish a user's identity by selecting from several built-in authentication methods, or by using a wizard to create your own custom authentication approach.



See Also:

["Establishing User Identity Through Authentication"](#)

About Controlling Access to Controls and Components

While conditions control the rendering and processing of specific controls or components on a page, authorization schemes control user access. Authorization is a broad term for controlling access to resources based on user privileges.

Authorization schemes use the identities established by authentication to grant privileges on applications and objects within them. You can specify an authorization scheme for an entire application, a page, or a specific page control such as a region, item, or button. For example, you could use an authorization scheme to selectively determine which tabs, regions, or navigation bar entries a user sees.



See Also:

["Providing Security Through Authorization"](#)

About Enabling Support for Bookmarks

Developers can assist users who bookmark application pages by either using zero as the session ID, or enabling Rejoin Sessions.

In previous releases, if the pages within an application were public and did not require authentication, you could assist users in bookmarking pages by using zero as the session ID. Although this approach is still supported, Oracle Application Express now includes a new feature called Rejoin Sessions that enables developers to control this type of support at the page, application, or instance-level.



See Also:

["About Rejoin Sessions"](#)

Understanding Session State Management

Oracle Application Express transparently maintains session state and provides developers with the ability to get and set session state values from any page in the application.

- [What is Session State?](#)
Session state enables developers to store and retrieve values for a user as the user navigates between different application pages.
- [About Session IDs](#)
A **session ID** is a unique number assigned a specific user for the duration of that user's visit (**session**)
- [What Is a Session?](#)
A **session** is a logical construct that establishes persistence (or stateful behavior) across page views. Each session is assigned a unique identifier. The Application Express engine uses this identifier (or session ID) to store and retrieve an application's working set of data (or session state) before and after each page view.

What is Session State?

Session state enables developers to store and retrieve values for a user as the user navigates between different application pages.

Hypertext Transfer Protocol (HTTP), the protocol over which HTML pages are most often delivered, is a stateless protocol. A web browser is only connected to the server for as long as it takes to download a complete page. Each page request is treated by the server as an independent event, unrelated to any page requests that happened previously or that may occur in the future. To access form values entered on one page on a subsequent page, the values must be stored as session state. Oracle Application Express transparently maintains session state and provides developers with the ability to get and set session state values from any page in the application.

About Session IDs

A **session ID** is a unique number assigned a specific user for the duration of that user's visit (**session**)

The Application Express engine establishes the identity of the user for each page request and the session ID to fetch session state from the database. The most visible location of the session ID is in the URL for a page request. The session ID displays as the third parameter in the URL, for example:

```
http://apex.somewhere.com/pls/apex/f?p=4350:1:220883407765693447
```

In this example, the session ID is 220883407765693447.

Another visible location is in the page's HTML POST data and indirectly in the contents of a session cookie. This cookie is sent by the Application Express engine during authentication and is maintained for the life of the application (or browser) session.

Oracle Application Express assigns new session IDs during authentication processing, records the authenticated user's identity with the session ID, and continually checks the session ID in each page request's URL or POST data with the session cookie and the session record in the database. These checks provide users with flexibility and security.

While the session ID is the key to session state, the session cookie and the session record safeguard the integrity of the session ID and the authentication status of the user.



See Also:

["About the Application Page URL"](#)

What Is a Session?

A **session** is a logical construct that establishes persistence (or stateful behavior) across page views. Each session is assigned a unique identifier. The Application Express engine uses this identifier (or session ID) to store and retrieve an application's working set of data (or session state) before and after each page view.

Because sessions are entirely independent of one another, any number of sessions can exist in the database at the same time. A user can also run multiple instances of an application simultaneously in different browser programs.

Sessions are logically and physically distinct from Oracle database sessions used to service page requests. A user runs an application in a single Oracle Application Express session from sign in to sign out with a typical duration measured in minutes or hours. Each page requested during that session results in the Application Express engine creating or reusing an Oracle database session to access database resources. Often these database sessions last just a fraction of a second.



See Also:

["Viewing Active Sessions" in *Oracle Application Express Administration Guide*](#)

Viewing Session State

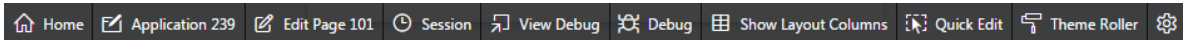
View the Session page by clicking **Session** on the Developer toolbar.

The Session page provides valuable information about the session in which the application is currently running. Most of the behavior of an Oracle Application Express application is driven by values in session state. For example, a button may display

conditionally based on the value of an item session state. This section describes how to view session state for a page.

To view the Session page:

1. Run the application as a developer.
2. Locate the Developer toolbar. By default, the Developer toolbar displays at the bottom of the page.



3. Click **Session** on the Developer toolbar.

The Session page appears and displays the following information displays at the top of the page.

- Application - Identifies the application name.
 - Session - Displays the session ID for the current session.
 - User - Identifies the current user.
 - Workspace - Displays the current workspace ID.
 - Browser Language - Identifies the current browser language.
4. Use the fields at the top of the page to alter the display.
 5. **Page** - To locate a page, enter the page number and click **Set**:
 6. **Find** - Enter a case insensitive query for and click **Set**. Use Find to search item name and item value, or collection name (depending upon your view).
 7. **Rows** - Select the number of items to display and click **Set**. Use Find to search item name and item value, or collection name (depending upon your view).
 8. **View** - Select a session state report to view and click **Set**.

 **Tip:**

Use Find to search item name and item value, or collection name (depending upon your view).

View options include:

- **Page Items** - Displays attributes of the page item, including the application ID, page number, item name, how the item displays (for example, check box, date picker, display only, text field, hidden, popup, radio group, and so on), the item value in session state, status of the session state (for example, Inserted, Updated, or Reset) and if the item is encrypted.
- **Application Items** - Displays application items. Application items are items that do not reside on a page. Application items are session state variables without the associated user interface properties.
- **Session State** - Displays session state details, including the application ID, page number, item name, the item value in session state, how the item displays (for example, check box, date picker, display only, text field, hidden, popup, radio group, and so on), label, encryption status, and data item ID.

- **Collections** - Displays currently defined collections.
- **All** - Displays page items, application items, session state, and collections.

 **See Also:**

- ["Understanding Page-Level Items"](#)
- ["Managing Application-Level Items"](#)
- ["About Using Collections"](#)
- ["Using Substitution Strings"](#) for information about referencing item values

Understanding URL Syntax

Use f?p syntax to link pages or call pages using a page alias or button URL.

- [About the Application Page URL](#)
The URL for each application page indicates the location of Oracle Application Express and identifies the address of Oracle Application Express, the application ID, page number, and session ID.
- [About Using f?p Syntax to Link Pages](#)
Create links between application page using f?p syntax.
- [Calling a Page Using an Application Alias or Page Alias](#)
Use URL syntax to call a page using an application alias and/or page alias.
- [Calling a Page from a Button URL](#)
Use URL syntax to call a page from a button URL.

About the Application Page URL

The URL for each application page indicates the location of Oracle Application Express and identifies the address of Oracle Application Express, the application ID, page number, and session ID.

Consider the following example:

```
http://apex.somewhere.com/pls/apex/f?p=4350:1:220883407765693447
```

This example indicates:

- `http:` is the scheme.
- `apex.somewhere.com` is the domain name of the server. It can also include a port number or an IP address.
- `pls` is the indicator to use the `mod_plsql` cartridge (if applicable)
- `apex` is the Database Access Descriptor (DAD) name. The DAD describes how HTTP Server connects to the database server so that it can fulfill an HTTP request. The default value is `apex`.
- `f?p=` is a prefix used by Oracle Application Express to route the request to the correct engine process.

- 4350 is the ID of the application being called. The application ID is a unique number that identifies each application.
- 1 is the number of the page within the application.
- 220883407765693447 is the session number. When you run an application, the Application Express engine generates a session number that serves as a key to the user's session state.



See Also:

"[Understanding Session State Management](#)" and "[About Publishing the Database Application URL](#)"

About Using f?p Syntax to Link Pages

Create links between application page using f?p syntax.

App Builder includes many wizards that automatically create these references for you. However, you may have to create the syntax yourself in some situations.

To create links between pages in an application, use the following f?p syntax:

```
f?p=App:Page:Session:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly
```

The following table describes the arguments you can pass when using f?p syntax.

Table 2-1 f?p Syntax Arguments

| Syntax | Description |
|---------|---|
| App | Indicates an application ID or alphanumeric application alias. An application alias is a unique alternate alphanumeric application identifier that developers can use in place of the application ID. You define an Application Alias editing the Application Definition. See Also: " Managing Application Attributes " |
| Page | Indicates a page number or alphanumeric page alias. A page alias is a unique alternate alphanumeric for a page number. Developers can use a page alias in place of the page number ID. You define a Page Alias by editing the page attributes. See Also: " Managing Page Attributes " |
| Session | Identifies a session ID. You can reference a session ID to create hypertext links to other pages that maintain the same session state by passing the session number. You can reference the session ID using the syntax: <ul style="list-style-type: none"> • Short substitution string: &SESSION. • PL/SQL: V('SESSION') • Bind variable: :APP_SESSION |

Table 2-1 (Cont.) f?p Syntax Arguments

| Syntax | Description |
|------------|--|
| Request | <p>Sets the value of REQUEST. Each application button sets the value of REQUEST to the name of the button which enables accept processing to reference the name of the button when a user clicks it. You can reference REQUEST using the syntax:</p> <ul style="list-style-type: none"> • Substitution string: &REQUEST. • PL/SQL: V('REQUEST') • Bind variable: :REQUEST |
| Debug | <p>Displays application processing details. Valid values for the DEBUG flag include:</p> <ul style="list-style-type: none"> • YES. • LEVELn • NO <p>Setting this flag to YES displays details about application processing. Setting this flag to LEVELn (where n is between 1 and 9) controls the level of debug detail that displays. The value of YES equals LEVEL4. You can reference the Debug flag using the following syntax:</p> <ul style="list-style-type: none"> • Short substitution string: &DEBUG. • PL/SQL: V('DEBUG') • Bind variable: :DEBUG <p>See Also: "Debugging an Application"</p> |
| ClearCache | <p>Clears the cache. This sets the value of items to null.</p> <p>To clear cached items on a single page, specify the numeric page number. To clear cached items on multiple pages, use a comma-separated list of page numbers. Clearing a page's cache also resets any stateful processes on the page. Individual or comma-separated values can also include collection names to be reset or the keyword RP, which resets region pagination on the requested page. The keyword APP clears cache for all pages and all application-level items in the current application and removes sort preferences for the current user. The keyword SESSION achieves the same result as the APP keyword, but clears items associated with all applications that have been used in the current session.</p> <p>See Also: "Clearing Session State"</p> |
| itemNames | Comma-delimited list of item names used to set session state with a URL. |
| itemValues | <p>List of item values used to set session state within a URL. To pass a comma in an item value, enclose the characters with backslashes. For example:</p> <pre>\123,45\</pre> <p>Every character sequence except backslash comma (\,) can be enclosed with backslash.</p> |

Table 2-1 (Cont.) f?p Syntax Arguments

| Syntax | Description |
|-----------------|---|
| PrinterFriendly | <p>Determines if the page is being rendered in printer friendly mode. If PrinterFriendly is set to Yes, then the page is rendered in printer friendly mode. The value of PrinterFriendly can be used in rendering conditions to remove elements such as regions from the page to optimize printed output.</p> <p>You can reference the printer friendly preference by using the following syntax:</p> <pre>V('PRINTER_FRIENDLY')</pre> <p>When referenced, the page displays using a printer friendly template. The Application Express engine displays all text within HTML form fields as text. The printer friendly template does not need to have the #FORM_OPEN# or #FORM_CLOSE# tags. The objective is to be able to display information with few tables and in a format suitable for printing.</p> |

Calling a Page Using an Application Alias or Page Alias

Use URL syntax to call a page using an application alias and/or page alias.

Application and page aliases must consist of valid Oracle identifiers, cannot contain any whitespace, and cannot be case-sensitive. The following example calls a page from within an application using an application alias and a page alias. This example runs the page *home* of the application *myapp* and uses the current session ID.

```
f?p=myapp:home:&APP_SESSION.
```

Application aliases must be unique within a workspace. If an application in a different workspace has the same application alias, use the `&c` argument to specify the workspace name. For example:

```
f?p=common_alias:home:&APP_SESSION.&c=WORKSPACE_A
```

Calling a Page from a Button URL

Use URL syntax to call a page from a button URL.

When you create a button, you can specify a URL to redirect to when the user clicks the button. This example runs page 6001 of application 6000 and uses the current session ID.

```
f?p=6000:6001:&APP_SESSION.
```

Note that this is only one approach to using a button. This method bypasses page submission and acts as a hyperlink on the page. You can also have a button click submit a page. In that approach, clicking the button submits the page for processing, allowing forms to be submitted and session state to be saved.



See Also:

["Managing Buttons"](#) and ["APP_SESSION"](#)

Managing Session State Values

Manage session state to store and retrieve values for a user as the user navigates between different application pages.

When building interactive, data-driven web applications, the ability to access and manage session state values is critical. In Oracle Application Express, session state is automatically managed for every page and easily referenced in static HTML or logic controls such as processes or validations.

- [About Referencing Session State](#)
Reference item values stored in session state in regions, computations, processes, validations, and branches. An item can be a field, a text area, a password, a select list, or a check box.
- [About Setting Session State](#)
Set session state using form submissions, bind variables, computations, or f?p syntax.
- [Clearing Session State](#)
Clearing a cached value resets the value to null. You can clear the cached value for specific items, all items on a page, all pages in an application, or the current user session.
- [Referencing Session State Using Bind Variable Syntax](#)
Use bind variable syntax anywhere where you are using SQL or PL/SQL to reference session state of a specified item.
- [About Session Cloning](#)
Use session cloning to open a new browser window in an Oracle Application Express application and have it generate a new distinct session identifier and copy the session values from the original Application Express session to the new one.



See Also:

["Understanding Session State Management"](#) and ["Referencing Item Values"](#)

About Referencing Session State

Reference item values stored in session state in regions, computations, processes, validations, and branches. An item can be a field, a text area, a password, a select list, or a check box.

The following table describes the supported syntax for referencing item values.

Table 2-2 Syntax for Referencing Item Values

| Type | Syntax | Description |
|---------------------|--|--|
| SQL | Standard item syntax: :MY_ITEM Syntax for items containing special characters: :"MY_ITEM" | For items whose names are no longer than 30 characters, precede the item name with a colon (:). Use this syntax for references within a SQL query and within PL/SQL. To reference page items containing special, multibyte, or unicode characters, wrap the page item name in double quotation marks. |
| PL/SQL | V('MY_ITEM') | Use PL/SQL syntax to reference an item value using the V function. You can use the shorthand, V function, in place of APEX_UTIL.GET_SESSION_STATE. Use this syntax when utilizing Oracle Application Express variables directly within an Oracle database object, such as a function, trigger, or Oracle Data Redaction policy. See Also: <i>Oracle Application Express API Reference</i> |
| PL/SQL | NV('MY_NUMERIC_ITEM') | Use standard PL/SQL syntax referencing the numeric item value using the NV function. You can use the shorthand, NV function, in place of APEX_UTIL.GET_NUMERIC_SESSION_STATE. See Also: <i>Oracle Application Express API Reference</i> |
| Static text (exact) | Standard item syntax: &MY_ITEM. Syntax for items containing special characters: &"MY_ITEM". | For static text or an exact substitution, use the convention &ITEM_NAME followed by a period (.). To reference page items containing special, multibyte, or unicode characters, wrap the page item name in double quotation marks. |

About Setting Session State

Set session state using form submissions, bind variables, computations, or f?p syntax.

You can set the value of an item in your application and therefore set session state using the following methods:

- Form submission. See the next section.
- Bind variable. See "[Referencing Session State Using Bind Variable Syntax.](#)"
- Computation. See "[Understanding Application Computations.](#)"
- f?p syntax. See "[About Using f?p Syntax to Link Pages.](#)"

About Setting Session State with a Form Submission

When a user submits a page, the Application Express engine automatically stores values typed into fields (items) in session state. For example, suppose you have an application containing two pages. The first page of the application contains a form in which a user can enter a phone number. You defined this form by creating an item named `P1_PHONENO`. On the second page, you want to display the information the user enters in the form.

When the page is submitted, Oracle Application Express captures the value entered in the phone number field and stores the value for future use. On the second page, the

phone number entered by the user can then be retrieved from session state using the name `P1_PHONE_NO` with the appropriate syntax from [Table 2-2](#).

Clearing Session State

Clearing a cached value resets the value to null. You can clear the cached value for specific items, all items on a page, all pages in an application, or the current user session.

- [About Clearing Cache for an Item](#)
- [About Clearing Cache for All Page Items](#)
- [Clearing Cache for an Entire Application](#)
- [About Resetting an Application Completely](#)
- [About Clearing Cache for the Current User Session](#)

About Clearing Cache for an Item

Clearing cache for a single item resets the value of the item to null. For example, you might use this approach to make sure a specific item's value is null when a page is prepared for rendering.

Example 2-1 Example: Clearing Cache for an Item

The following example uses standard `f?p` syntax to clear the cache for an item. This example calls page 5 of application 100. Placing `MY_ITEM` in the `ClearCache` position of the `f?p` syntax resets the value of `MY_ITEM` to `NULL`.

```
f?p=100:5:&APP_SESSION.::NO:MY_ITEM
```

The following example resets the value of the items `THE_EMPNO` and `THE_DEPTNO`:

```
f?p=100:5:&APP_SESSION.::NO:THE_EMPNO,THE_DEPTNO
```

About Clearing Cache for All Page Items

Caching application items is an effective way to maintain session state. However, there are occasions when you may want to clear the cache for all items on a page. For example, suppose you needed to clear all fields on a page when a user clicks a link that creates a new order. By clearing the cache for an entire page, you set the value of all items on the page to null.

Example 2-2 Example: Clearing Cache for Two Pages and Resetting Pagination

This example clears the session cache for two pages and resets pagination.

```
f?p=6000:6003:&APP_SESSION.::NO:RP,6004,6014
```

This example:

- Runs page 6003 of application 6000 and uses the current session ID.
- Indicates to not show debug information (`NO`).
- Clears all values maintained by the current session's cache for items of pages 6004 and 6014.
- Resets region pagination (`RP`) on page 6003 (the requested page).

Example 2-3 Example: Clearing Cache on a Page and Passing an Item Value

This example shows how to implement an update form. It clears existing information and sets the item's value (typically a primary key).

```
f?p=6000:6003:&APP_SESSION.::NO:6003:MY_ITEM:1234
```

This example:

- Runs page 6003 of application 6000 and uses the current session ID
- Indicates to not show debug information (NO)
- Clears all values maintained by the current session's cache for items on page 6003
- Sets the session state of an item called `MY_ITEM` to the value 1234

Example 2-4 Example: Clearing Session Cache on a Page and Passing Values to Multiple Items

This example demonstrates how to implement an update form. It clears existing information, sets the item's value (typically a primary key), and passes values to multiple items.

```
f?p=6000:6004:&APP_SESSION.::NO:6003:MY_ITEM1,MY_ITEM2,MY_ITEM3:1234,,5678
```

This example:

- Runs page 6004 of application 6000 and uses the current session ID
- Clears the current session's cache for items on page 6003
- Indicates to not show debug information (NO)
- Sets the value of `MY_ITEM1` to 1234, sets the value of `MY_ITEM2` to null (indicated by the comma used as placeholder), and sets the value of `MY_ITEM3` to 5678

Clearing Cache for an Entire Application

This example clears the application's cache by using `f?p` syntax and creating a `Clear Cache` argument using the keyword `APP`.

```
f?p=App:Page:Session::NO:APP
```

About Resetting an Application Completely

Resetting the cache for an entire application does not restore the application to a completely reset state. For example, if an application includes on-new instance computations or on-new instance processes, the Application Express engine runs these computations and processes when the application session is created. Then, it processes the clear cache request and displays the requested page.

To reset an application completely without a session ID (if no cookie is used to track the session ID), you must request it using a URL without a session ID, or by calling `APEX_UTIL.CLEAR_APP_CACHE` from another application. If the session ID is tracked using a cookie, you must logout to reset the state.

About Clearing Cache for the Current User Session

Another approach to clearing an application's cache is to create a `Clear Cache` argument using the keyword `SESSION`. For example:

```
f?p=6000:6004:12507785108488427528::NO:SESSION
```

Referencing Session State Using Bind Variable Syntax

Use bind variable syntax anywhere where you are using SQL or PL/SQL to reference session state of a specified item.

- [About Using Bind Variable Syntax](#)
- [About Using Bind Variables in Regions Based on a SQL Query or LOV](#)
- [About Using Bind Variables in Regions Based on PL/SQL](#)

About Using Bind Variable Syntax

In the following example, the search string is a page item.

```
SELECT * FROM employees WHERE last_name like '%' || :SEARCH_STRING || '%'
```

If the region type is defined as SQL Query, you can reference the value using standard SQL bind variable syntax. Using bind variables ensures that parsed representations of SQL queries are reused by the database, optimizing memory usage by the server.

When using bind variable syntax, remember the following rules:

- Bind variable names must correspond to an item name.
- Bind variable names are not case-sensitive.
- Bind variable names cannot be longer than 30 characters (that is, they must be a valid Oracle identifier).

Although page item and application item names can be up to 255 characters, if you intend to use an application item within SQL using bind variable syntax, the item name must be 30 characters or less.

About Using Bind Variables in Regions Based on a SQL Query or LOV

If your region type is defined as a SQL Query, SQL Query (plsqli function body returning SQL query), or list of values (LOV), you can reference session state using the following syntax:

```
:MY_ITEM
```

One common way to do this is to incorporate a session state variable in a `WHERE` clause. The following example shows how to bind the value of the item `THE_DEPTNO` into a region defined from a SQL Query.

```
SELECT last_name, job_id, salary
FROM employees
WHERE department_id = :THE_DEPTNO
```

**See Also:**

"[About Regions](#)" for information about creating regions

About Using Bind Variables in Regions Based on PL/SQL

For region types, processes, computations, validations, conditions, and so on that are defined as PL/SQL dynamic content, regions are constructed using PL/SQL anonymous block syntax. In other words, the beginning and ending keywords are used to enclose the PL/SQL block. For example:

```
IF :P1_JOB IS NOT NULL THEN
  INSERT INTO employees (employee_id, first_name, job_id)
  VALUES (:P1_EMP_ID, :P1_NAME, :P1_JOB)
end if;
```

In this example, the values of the `employee_id`, `first_name`, and `job_id` are populated by the values of `P1_EMP_ID`, `P1_NAME`, and `P1_JOB`.

About Session Cloning

Use session cloning to open a new browser window in an Oracle Application Express application and have it generate a new distinct session identifier and copy the session values from the original Application Express session to the new one.

In previous releases, opening multiple windows (or browser tabs) in the same Oracle Application Express application resulted in a number of issues. Typically, all the browser tabs shared the same session and session state which resulted in unpredictable and undesirable results.

To use session cloning, the developer must provide a method for end user to open a new browser tab and specify the `REQUEST` value of `APEX_CLONE_SESSION`. The following is an example URL:

```
f?p=&APP_ID. :&APP_PAGE_ID. :&APP_SESSION. :APEX_CLONE_SESSION
```

Best Practices When Using Session Cloning

When using this session cloning, developers should remember:

- The idle time associated with the Application Express session would be affected by any of the Application Express sessions, original or cloned ones.
- When a user logs out of one session (original or cloned), all other associated sessions will be logged out.
- The maximum session duration would be a function of the original Application Express session and when it was created.
- An administrative user can enable or disable this feature using the procedure:

```
apex_instance_admin.set_parameter(
  p_parameter => 'CLONE_SESSION_ENABLED',
  p_value     => 'Y');
```

**See Also:**

Oracle Application Express API Reference

Using Substitution Strings

Use substitution strings within a page template or region source to pass information and replace a character string with another value.

- [About Using Substitution Strings](#)
- [Controlling Output Escaping in Substitution Strings](#)
- [About Determining Substitution String Usage within Templates](#)
- [Using Built-in Substitution Strings](#)

About Using Substitution Strings

You can use substitution strings in App Builder in the following ways:

- Include a substitution string within a template to reference component values
- Reference page or application items using `&ITEM.` syntax
- Use built-in substitution strings

Substitution Strings within Templates

Special substitution strings available within a template are denoted by the number symbol (#). For example:

```
#ABC#
```

**Tip:**

You can also use JavaScript API `apex.util.applyTemplate` to evaluate templates on the client side. See [Oracle Application Express JavaScript API Reference](#).

Substitution Strings for Page or Application Items

To reference page or application items using substitution variables:

1. Reference the page or application item in all capital letters.
2. Precede the item name with an ampersand (&).
3. Append a period (.) to the item name.

For example, you would refer to an application item named `F101_X` in a region, a region title, an item label, or in any of numerous other contexts in which static text is used, for example:

```
&F101_X.
```

**Tip:**

Notice the required trailing period. When the page is rendered, Application Express engine replaces the value the substitution string with the value of the item `F101_X`.

Substitution Strings for Interactive Grid Columns

To reference columns in an interactive grid using substitution variables:

1. Reference the column name in all capital letters.
2. Precede the column name with an ampersand (&).
3. Append a period (.) to the item name.

For example, to reference the interactive grid column `ENAME`:

```
&ENAME.
```

Substitution Strings for Other Report Columns

To reference columns in other reports using substitution variables:

1. Reference the column name in all capital letters.
2. Precede the column name with a pound sign (#).
3. Append a pound sign (#) to the item name.

For example, to reference the interactive or classic report column `ENAME`:

```
#ENAME#
```

Using Double Quotation Marks for Special Characters or Case Sensitivity

Enclose items or columns with double quotation marks if the name contains special characters or is case sensitive. For example:

```
&"ITEM-OR-COLUMN-NAME".
```

Use the following syntax:

```
&"<name>" [!<format>].
```

Where:

- "`<name>`" is an application item or page item or column name. If not quoted, then it must be in set `A-Z-0-9_$$`
- "`<name>`" is `APP_TEXT$<message-name>`

Where `<message-name>` is an Application Express text message name.

- `<format>` is a predefined filter name: `HTML`, `ATTR`, `JS`, `RAW`, `STRIPHTML` as described in ["Controlling Output Escaping in Substitution Strings."](#)

Controlling Output Escaping in Substitution Strings

You can escape special characters in the substitution value by appending an exclamation mark (!) followed by a predefined filter name to a page or application item name, report column, or other substitution string. Output escaping is an important security technique to avoid Cross Site Scripting (XSS) attacks in the browser. Oracle Application Express already makes a best effort to automatically escape characters in a HTML or JavaScript context. With this extended syntax, developers have fine grained control over the output.

Available Escape Filters

Available escape filters include:

- HTML escapes reserved HTML characters, for example:
 - Item:
`&P1_DEPTNO!HTML.`
 - Interactive grid column:
`&ENAME!HTML.`
 - Columns in other reports:
`#ENAME!HTML#`

See Also:

"APEX_ESCAPE.HTML" in *Oracle Application Express API Reference*

- ATTR escapes reserved characters in a HTML attribute context, for example:
 - Interactive grid column:
`&ENAME!ATTR.`
 - Columns in other reports:
`#ENAME!ATTR#`

See Also:

"APEX_ESCAPE.HTML_ATTRIBUTE" in *Oracle Application Express API Reference*

- JS escapes reserved characters in a JavaScript context, for example:
 - Item:
`&P1_DEPTNO!JS.`
 - Interactive grid column:
`&ENAME!JS.`
 - Columns in other reports:


```
#ENAME!JS#
```

 **See Also:**

"APEX_ESCAPE.JS_LITERAL" in *Oracle Application Express API Reference*

- RAW preserves the original item value and does not escape characters, for example:
 - Item:
`&P1_DEPTNO!RAW.`
 - Interactive grid column:
`&ENAME!RAW.`
 - Columns in other reports:
`#ENAME!RAW#`
- STRIPHTML removes HTML tags from the output and escapes reserved HTML characters, for example:
 - Item:
`&P1_DEPTNO!STRIPHTML.`
 - Interactive grid column:
`&ENAME!STRIPHTML.`
 - Columns in other reports:
`#ENAME!STRIPHTML#`

About Determining Substitution String Usage within Templates

You can determine what template-specific substitution strings are supported in which templates by viewing the template definition. See "[Creating Custom Themes](#)."

Using Built-in Substitution Strings

App Builder supports many built-in substitution strings. You can reference these substitution strings to achieve specific types of functionality.

 **Tip:**

Note that bind variable `:USER` has special meaning within the database. Also, the term **Direct PL/SQL** refers to PL/SQL that can be used in stored database objects such as procedures and functions.

- [APEX\\$ROW_NUM](#)
- [APEX\\$ROW_SELECTOR](#)

- APEX\$ROW_STATUS
- APP_ID
- APP_ALIAS
- APP_AJAX_X01, ... APP_AJAX_X10
- APP_BUILDER_SESSION
- APP_DATE_TIME_FORMAT
- APP_IMAGES
- APP-NLS_DATE_FORMAT
- APP-NLS_TIMESTAMP_FORMAT
- APP-NLS_TIMESTAMP_TZ_FORMAT
- APP_PAGE_ALIAS
- APP_PAGE_ID
- APP_REQUEST_DATA_HASH
- APP_SESSION
- APP_SESSION_VISIBLE
- APP_TITLE
- APP_UNIQUE_PAGE_ID
- APP_USER
- AUTHENTICATED_URL_PREFIX
- BROWSER_LANGUAGE
- CURRENT_PARENT_TAB_TEXT
- DEBUG
- HOME_LINK
- IMAGE_PREFIX
- JET_BASE_DIRECTORY
- JET_CSS_DIRECTORY
- JET_JS_DIRECTORY
- LOGIN_URL
- LOGOUT_URL
- APP_TEXT\$Message_Name, APP_TEXT\$Message_Name\$Lang
- PRINTER_FRIENDLY
- PROXY_SERVER
- PUBLIC_URL_PREFIX
- REQUEST
- Using REQUEST
- SCHEMA OWNER
- SQLERRM

- `SYSDATE_YYYYMMDD`
- `THEME_DB_IMAGES`
- `THEME_IMAGES`
- `WORKSPACE_IMAGES`
- `WORKSPACE_ID`



See Also:

"Substitutions" and "Establishing User Identity Through Authentication"

APEX\$ROW_NUM

`APEX$ROW_NUM` refers the currently processed row number of a submitted tabular form data. You can use this placeholder in validations, processes, and conditions associated with a tabular form to refer to the row number of the currently processed tabular form row. [Table 2-3](#) describes the supported syntax for referencing `APEX$ROW_NUM`.

Table 2-3 APEX\$ROW_NUM

| Reference Type | Syntax |
|---------------------|----------------------|
| Bind variable | :APEX\$ROW_NUM |
| PL/SQL | V('APEX\$ROW_NUM') |
| Substitution string | &APEX\$ROW_NUM. |

APEX\$ROW_SELECTOR

Use `APEX$ROW_SELECTOR` in validations, processes, and conditions associated with a tabular form to refer to the row selector check box in a tabular form. This placeholder returns `X` if the tabular form row selector check box of the currently processed tabular form row is checked and `NULL` if it unchecked. [Table 2-4](#) describes the supported syntax for referencing `APEX$ROW_SELECTOR`.

Table 2-4 APEX\$ROW_SELECTOR

| Reference Type | Syntax |
|---------------------|---------------------------|
| Bind variable | :APEX\$ROW_SELECTOR |
| PL/SQL | V('APEX\$ROW_SELECTOR') |
| Substitution string | &APEX\$ROW_SELECTOR. |

APEX\$ROW_STATUS

Use `APEX$STATUS` in validations, processes, and conditions associated with a tabular form to refer to the row status in a tabular form. This placeholder returns the status of c

if created, U if updated, or D if deleted for the currently processed tabular form row. [Table 2-5](#) describes the supported syntax for referencing `APEX$ROW_STATUS`.

Table 2-5 APEX\$ROW_STATUS

| Reference Type | Syntax |
|---------------------|-----------------------|
| Bind variable | :APEX\$ROW_STATUS |
| PL/SQL | V('APEX\$ROW_STATUS') |
| Substitution string | &APEX\$ROW_STATUS. |

APP_ID

`APP_ID` identifies the application ID of the currently executing application. [Table 2-6](#) describes the supported syntax for referencing `APP_ID`.

Table 2-6 APP_ID Syntax

| Reference Type | Syntax |
|---------------------|---------------------------------------|
| Bind variable | :APP_ID |
| Direct PL/SQL | APEX_APPLICATION.G_FLOW_ID (A NUMBER) |
| PL/SQL | NV('APP_ID') |
| Substitution string | &APP_ID. |

The following is an example of a substitution string reference:

```
f?p=&APP_ID.:40:&APP_SESSION.
```

APP_ALIAS

`APP_ALIAS` is an alphanumeric name for the current application. `APP_ALIAS` is different from the `APP_ID` in that the `APP_ID` must be unique over all workspaces and all applications hosted in one database. In contrast, `APP_ALIAS` must be unique within a workspace. For example, by using the same `APP_ALIAS` you can create the application, ABC, in two different workspaces. You can use `APP_ALIAS` almost anywhere `APP_ID` can be used. For example, `f?p` syntax can use an `APP_ALIAS` or an application ID as demonstrated in this example:

```
f?p=ABC:1:&APP_SESSION.
```

This example runs application ABC, page 1 using the current session.

[Table 2-7](#) describes the supported syntax for referencing `APP_ALIAS`.

Table 2-7 APP_ALIAS Syntax

| Reference Type | Syntax |
|----------------|----------------|
| Bind variable | :APP_ALIAS |
| PL/SQL | V('APP_ALIAS') |

Table 2-7 (Cont.) APP_ALIAS Syntax

| Reference Type | Syntax |
|---------------------|-------------|
| Substitution string | &APP_ALIAS. |

The following is an HTML example:

```
Click me to go to page 1 <a href="f?p=&APP_ALIAS.:1:&APP_SESSION."> of the current application</a>
```

APP_AJAX_X01, ... APP_AJAX_X10

APP_AJAX_Xnn specifies the values of the APP_AJAX_X01, ... APP_AJAX_X10 URL parameters most recently passed to or set within the show or accept modules. You typically use these variables in On Demand AJAX processes.

APP_AJAX_Xnn describes the supported syntax for referencing APP_AJAX_Xnn.

Table 2-8 APP_AJAX_Xnn Syntax

| Reference Type | Syntax |
|---------------------|---|
| Bind variable | ... :APP_AJAX_X01, ... :APP_AJAX_X10 |
| PL/SQL | ... v('APP_AJAX_X01'), ... v('APP_AJAX_X10') |
| Substitution string | ... &APP_AJAX_X01., ... &APP_AJAX_X10. |

See Also:

G_X01, ... G_X10 variables in the APEX_APPLICATION in Oracle Application Express API Reference

APP_BUILDER_SESSION

If the user is also logged in to the workspace as a developer, APP_BUILDER_SESSION contains the current session ID of the development environment. Otherwise, APP_BUILDER_SESSION is null.

Table 2-9 APP_BUILDER_SESSION Syntax

| Reference Type | Syntax |
|---------------------|--------------------------|
| Bind variable | :APP_BUILDER_SESSION |
| PL/SQL | V('APP_BUILDER_SESSION') |
| Substitution string | &APP_BUILDER_SESSION. |

APP_DATE_TIME_FORMAT

APP_DATE_TIME_FORMAT is the application date time format of the application. This value reflects the format specified in the Application Date Time Format attribute of the Globalization settings of an application. If the Application Date Time Format is not set in an application, then a reference to APP_DATE_TIME_FORMAT returns the database session NLS date format and the NLS time format. [Table 2-10](#) describes the supported syntax for referencing APP_DATE_TIME_FORMAT.

Table 2-10 APP_DATE_TIME_FORMAT Syntax

| Reference Type | Syntax |
|---------------------|---------------------------|
| Bind variable | :APP_DATE_TIME_FORMAT |
| PL/SQL | V('APP_DATE_TIME_FORMAT') |
| Substitution string | &APP_DATE_TIME_FORMAT. |

APP_IMAGES

Use this substitution string to reference uploaded images, JavaScript, and cascading style sheets that are specific to a given application and are not shared over many applications. If you upload a file and make it specific to an application, then you must use this substitution string, or bind variable. [Table 2-11](#) describes the supported syntax for referencing APP_IMAGES.

Table 2-11 APP_IMAGES Syntax

| Reference Type | Syntax |
|-----------------------|-----------------|
| Bind variable | :APP_IMAGES |
| Direct PL/SQL | Not available. |
| PL/SQL | V('APP_IMAGES') |
| Substitution string | &APP_IMAGES. |
| Template substitution | #APP_IMAGES# |



See Also:

"IMAGE_PREFIX," "WORKSPACE_IMAGES," and "About Managing Images"

APP_NLS_DATE_FORMAT

APP_NLS_DATE_FORMAT is the application date format of the database session. This value reflects the format specified in the Application Date Format attribute of the Globalization settings of the application. However, if the Application Date Format is not set, then APP_NLS_DATE_FORMAT returns the NLS_DATE_FORMAT value of the database

session at the start of the request to the Application Express engine. [Table 2-12](#) describes the supported syntax for referencing `APP_NLS_DATE_FORMAT`.

Table 2-12 APP_NLS_DATE_FORMAT Syntax

| Reference Type | Syntax |
|---------------------|--------------------------|
| Bind variable | :APP_NLS_DATE_FORMAT |
| PL/SQL | V('APP_NLS_DATE_FORMAT') |
| Substitution string | &APP_NLS_DATE_FORMAT. |

APP_NLS_TIMESTAMP_FORMAT

`APP_NLS_TIMESTAMP_FORMAT` is the application timestamp format of the database session. This value reflects the format specified in the Application Timestamp Format attribute of the Globalization settings of the application. However, if the Application Timestamp Format is not set, then `APP_NLS_TIMESTAMP_FORMAT` return the `NLS_TIMESTAMP_FORMAT` value of the database session at the start of the request to the Application Express engine. [Table 2-13](#) describes the supported syntax for referencing `APP_NLS_TIMESTAMP_FORMAT`.

Table 2-13 APP_NLS_TIMESTAMP_FORMAT Syntax

| Reference Type | Syntax |
|---------------------|-------------------------------|
| Bind variable | :APP_NLS_TIMESTAMP_FORMAT |
| PL/SQL | V('APP_NLS_TIMESTAMP_FORMAT') |
| Substitution string | &APP_NLS_TIMESTAMP_FORMAT. |

APP_NLS_TIMESTAMP_TZ_FORMAT

`APP_NLS_TIMESTAMP_TZ_FORMAT` is the application timestamp time zone format of the database session. This value reflects the format specified in the Application Timestamp Time Zone Format attribute of the Globalization settings of an application. However, if the Application Timestamp Time Zone Format is not set, then `APP_NLS_TIMESTAMP_TZ_FORMAT` returns the `NLS_TIMESTAMP_TZ_FORMAT` value of the database session at the start of the request to the Application Express engine. [Table 2-14](#) describes the supported syntax for referencing `APP_NLS_TIMESTAMP_TZ_FORMAT`.

Table 2-14 APP_NLS_TIMESTAMP_TZ_FORMAT Syntax

| Reference Type | Syntax |
|---------------------|----------------------------------|
| Bind variable | :APP_NLS_TIMESTAMP_TZ_FORMAT |
| PL/SQL | V('APP_NLS_TIMESTAMP_TZ_FORMAT') |
| Substitution string | &APP_NLS_TIMESTAMP_TZ_FORMAT. |

APP_PAGE_ALIAS

APP_PAGE_ALIAS is an alphanumeric name for the current application page. A page alias is not case-sensitive and it is an optional page attribute. APP_PAGE_ALIAS is unique within an application. You can use APP_PAGE_ALIAS almost anywhere APP_PAGE_ID can be used. [Table 2-16](#) describes the supported syntax for referencing APP_PAGE_ID.

Table 2-15 APP_PAGE_ALIAS Syntax

| Reference Type | Syntax |
|---------------------|---------------------|
| Bind variable | :APP_PAGE_ALIAS |
| PL/SQL | V('APP_PAGE_ALIAS') |
| Substitution string | &APP_PAGE_ALIAS. |

The following is an HTML example:

The alias of the current page is: &APP_PAGE_ALIAS.

APP_PAGE_ID

APP_PAGE_ID is the current application page ID. For example, if your application was on page 3, then the result would be 3. Using this syntax is useful when writing application components that must work generically in multiple applications. [Table 2-16](#) describes the supported syntax for referencing APP_PAGE_ID.

Table 2-16 APP_PAGE_ID Syntax

| Reference Type | Syntax |
|----------------------|-------------------|
| Bind variable | :APP_PAGE_ID |
| PL/SQL | :APP_PAGE_ID |
| PL/SQL and Direct PL | NV('APP_PAGE_ID') |
| Substitution string | &APP_PAGE_ID. |

The following is an example of a substitution string reference:

```
f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.
```

APP_REQUEST_DATA_HASH

APP_REQUEST_DATA_HASH is a hash value of the request, item name, and item value parts in the URL. It is primarily useful to detect whether two browser requests passed different parameters to APEX. [Table 2-17](#) describes the supported syntax for referencing APP_REQUEST_DATA_HASH.

Table 2-17 APP_SESSION Syntax

| Reference Type | Syntax |
|---------------------|----------------------------|
| Bind variable | :APP_REQUEST_DATA_HASH |
| PL/SQL | V('APP_REQUEST_DATA_HASH') |
| Substitution string | &APP_REQUEST_DATA_HASH. |

APP_SESSION

APP_SESSION is the most commonly used built-in substitution strings. You can use this substitution string to create hypertext links between application pages that maintain a session state by passing the session number. Note that you can also use the substitution string SESSION in place of APP_SESSION. Table 2-18 describes the supported syntax for referencing APP_SESSION.

Table 2-18 APP_SESSION Syntax

| Reference Type | Syntax |
|----------------------|---|
| Bind variable | :APP_SESSION |
| PL/SQL | V('APP_SESSION') |
| Short PL/SQL | V('SESSION') |
| Substitution string | &APP_SESSION. |
| SYS_CONTEXT variable | SYS_CONTEXT('APEX\$SESSION', 'APP_SESSION') |

Consider the following examples:

- From within an HTML region:

```
<a href="f?p=100:5:&APP_SESSION.">click me</a>
```

- Using PL/SQL:

```
htf.anchor('f?p=100:5:'||V('APP_SESSION'),'click me');
```

- Using a SQL query:

```
SELECT htf.anchor('f?p=100:5:'||:APP_SESSION,'click me') FROM DUAL;
```

- Using the SYS_CONTEXT variable:

```
SELECT ... WHERE apex_session_id = SYS_CONTEXT('APEX$SESSION', 'APP_SESSION')
```

Oracle Application Express sets up the APEX\$SESSION context when it starts to process an incoming request. For example, you can use the value of 'APP_SESSION' to access the current application session in queries and VPD (Virtual Private Database) security policies that protect your table data.

APP_SESSION_VISIBLE

APP_SESSION_VISIBLE is similar to the built-in substitution APP_SESSION. Use this substitution string to create hypertext links between application pages that maintain a session state by passing the session number. APP_SESSION_VISIBLE always returns '0'

when users are not authenticated to an application and they are using the Zero Session ID feature of Oracle Application Express. [Table 2-19](#) describes the supported syntax for referencing `APP_SESSION_VISIBLE`.

Table 2-19 APP_SESSION_VISIBLE Syntax

| Reference Type | Syntax |
|---------------------|--------------------------|
| Bind variable | :APP_SESSION_VISIBLE |
| PL/SQL | V('APP_SESSION_VISIBLE') |
| Substitution string | &APP_SESSION_VISIBLE. |

Consider the following examples:

- From within an HTML region:

```
<a href="f?p=100:5:&APP_SESSION_VISIBLE.">click me</a>
```

- Using PL/SQL:

```
sys.htf.anchor('f?p=100:5:' || v('APP_SESSION_VISIBLE'), 'click me');
```

- Using a SQL query:

```
SELECT sys.htf.anchor('f?p=100:5:' || :APP_SESSION_VISIBLE, 'clickme') FROM DUAL;
```

APP_TITLE

`APP_TITLE` is an alphanumeric title for the current application. The title is derived from an application substitution string called `APP_TITLE`. If not defined the Logo attribute will be used if it is of type text. The last fallback is the application name. [Table 2-20](#) describes the supported syntax for referencing `APP_TITLE`.

Table 2-20 APP_TITLE Syntax

| Reference Type | Syntax |
|---------------------|----------------|
| Bind variable | :APP_TITLE |
| PL/SQL | v('APP_TITLE') |
| Substitution string | &APP_TITLE. |

The following is an HTML example:

```
The title of the current application is: &APP_TITLE!HTML.
```

APP_UNIQUE_PAGE_ID

`APP_UNIQUE_PAGE_ID` is an integer generated from an Oracle sequence which is unique for each page view. This number is used by applications to prevent duplicate page submissions and can be used for other purposes. For example, to make a unique URL and avoid browser caching issues, you can embed this number in the request or debug column in calls to the `f` procedure. [Table 2-21](#) describes the supported syntax for referencing `APP_UNIQUE_PAGE_ID`.

Table 2-21 APP_UNIQUE_PAGE_ID Syntax

| Reference Type | Syntax |
|---------------------|-------------------------|
| Bind variable | :APP_UNIQUE_PAGE_ID |
| PL/SQL | V('APP_UNIQUE_PAGE_ID') |
| Substitution string | &APP_UNIQUE_PAGE_ID. |

The following is an HTML example:

```
SELECT 'f?p=100:1:' || :APP_SESSION || ':' || :APP_UNIQUE_PAGE_ID ||
      ':::Pl_EMPNO:' || employee_id,
      first_name,
      job_id
FROM employees
```

Note the use of the `APP_UNIQUE_PAGE_ID` in the request column. This makes this URL unique and may avoid excessive browser caching problems.

APP_USER

`APP_USER` is the current user running the application. Depending upon your authentication model, the value of the user is set differently. If the application is running using database authentication, then the value of the user is the same as the database pseudo column `USER`. If the application uses an authentication scheme that requires the user to authenticate, the value of `APP_USER` is set by the authentication scheme, usually to the user name used during authentication. [Table 2-22](#) describes the supported syntax for referencing `APP_USER`.

Table 2-22 APP_USER Syntax

| Reference Type | Syntax |
|----------------------|--|
| Bind variable | :APP_USER |
| PL/SQL | V('APP_USER') |
| Substitution string | &APP_USER. |
| SYS_CONTEXT variable | SYS_CONTEXT('APEX\$SESSION', 'APP_USER') |

Consider the following examples:

- From within an HTML region:
Hello you are logged in as &APP_USER.
- Using PL/SQL:
`http.p('Hello you are logged in as' || V('APP_USER'));`
- As a bind variable:
`SELECT * FROM some_table WHERE user_id = :APP_USER`
- Using the `SYS_CONTEXT` variable:
`SELECT ... WHERE username = SYS_CONTEXT('APEX$SESSION', 'APP_USER')`

Oracle Application Express sets up the `APEX$SESSION` context when it starts to process an incoming request. For example, you can use the value of `'APP_USER'` to access the current application user in queries and VPD (Virtual Private Database) security policies that protect your table data.



See Also:

"[Authentication](#)" for information about the Public User attribute

AUTHENTICATED_URL_PREFIX

This application-level attribute identifies a valid authenticated prefix (that is, a logged in URL prefix). You can use a relative path or a full path beginning with `http`. This item is useful if your application can be run in both authenticated (logged in) and public (not logged in) modes. You can use `AUTHENTICATED_URL_PREFIX` to construct a link to an authenticated page. This item is most useful when using basic database authentication because changes to the URL can require authentication. [Table 2-23](#) describes the supported syntax for referencing `AUTHENTICATED_URL_PREFIX`.

Table 2-23 AUTHENTICATED_URL_PREFIX Syntax

| Reference Type | Syntax |
|---------------------|-------------------------------|
| Bind variable | :AUTHENTICATED_URL_PREFIX |
| PL/SQL | V('AUTHENTICATED_URL_PREFIX') |
| Substitution string | &AUTHENTICATED_URL_PREFIX. |

BROWSER_LANGUAGE

`BROWSER_LANGUAGE` refers to the web browser's current language preference. [Table 2-24](#) describes the supported syntax for referencing `BROWSER_LANGUAGE`.

Table 2-24 BROWSER_LANGUAGE Syntax

| Reference Type | Syntax |
|---------------------|-------------------------------------|
| Bind variable | :BROWSER_LANGUAGE |
| Direct PL/SQL | APEX_APPLICATION.G_BROWSER_LANGUAGE |
| PL/SQL | V('BROWSER_LANGUAGE') |
| Substitution string | &BROWSER_LANGUAGE. |

CURRENT_PARENT_TAB_TEXT

`CURRENT_PARENT_TAB_TEXT` is most useful in page templates, but is only relevant for applications that use two-level tabs (that is, parent and standard tabs). Use this string to reference the parent tab label. This substitution string enables you to repeat the currently selected parent tab within the page template. [Table 2-25](#) describes the supported syntax for referencing `CURRENT_PARENT_TAB_TEXT`.

Table 2-25 CURRENT_PARENT_TAB_TEXT Syntax

| Reference Type | Syntax |
|---------------------|---------------------------|
| Bind variable | Not Available. |
| Substitution string | &CURRENT_PARENT_TAB_TEXT. |

DEBUG

Valid values for the `DEBUG` flag are Yes or No. Turning debug on shows details about application processing. If you write your own custom code, you may want to generate debug information only if the debug mode is set to Yes. [Table 2-26](#) describes the supported syntax for referencing `DEBUG`.

Table 2-26 DEBUG Syntax

| Reference Type | Syntax |
|---------------------|--------------------------|
| Bind variable | :DEBUG |
| Direct PL/SQL | APEX_APPLICATION.G_DEBUG |
| PL/SQL | V('DEBUG') |
| Substitution string | &DEBUG. |

The following is an example of a substitution string reference that preserves the current value of `DEBUG`:

```
f?p=100:1:&APP_SESSION.::&DEBUG
```

HOME_LINK

`HOME_LINK` is the home page of an application. The Application Express engine redirects to this location if no page is given and if no alternative page is dictated by the authentication scheme's logic. You define the Home URL on the User Interface Details page.

[Table 2-27](#) describes the supported syntax for referencing `HOME_LINK`.

Table 2-27 HOME_LINK Syntax

| Reference Type | Syntax |
|---------------------|------------------------------|
| Direct PL/SQL | APEX_APPLICATION.G_HOME_LINK |
| PL/SQL | V('HOME_LINK') |
| Template Reference | #HOME_LINK# |
| Substitution String | &HOME_LINK. |

 **See Also:**

"[Managing the Application User Interface](#)" and "[User Interface Details Page](#)" for information about the Home URL attribute

IMAGE_PREFIX

The value of `IMAGE_PREFIX` determines the virtual path the web server uses to point to the images directory distributed with Oracle Application Express. To reference uploaded images, use `WORKSPACE_IMAGES` and `APP_IMAGES`. [Table 2-28](#) describes the supported syntax for referencing `IMAGE_PREFIX`.

Table 2-28 `IMAGE_PREFIX` Syntax

| Reference Type | Syntax |
|-----------------------|--|
| Bind variable | : <code>IMAGE_PREFIX</code> |
| Direct PL/SQL | <code>APEX_APPLICATION.G_IMAGE_PREFIX</code> |
| PL/SQL | <code>V('IMAGE_PREFIX')</code> |
| Substitution string | <code>&IMAGE_PREFIX.</code> |
| Template Substitution | <code>#IMAGE_PREFIX#</code> |

 **See Also:**

- "[APP_IMAGES](#)"
- "[WORKSPACE_IMAGES](#)"
- "[Editing Application Attributes](#)"

JET_BASE_DIRECTORY

Use the `JET_BASE_DIRECTORY` substitution string to reference the base directory of the Oracle JavaScript Extension Toolkit (JET) which ships with Oracle Application Express. Supported syntax for referencing `JET_BASE_DIRECTORY` :

```
#JET_BASE_DIRECTORY#
```

JET_CSS_DIRECTORY

Use the `JET_CSS_DIRECTORY` substitution string to reference the base directory of the Oracle JavaScript Extension Toolkit (JET) which ships with Oracle Application Express. Supported syntax for referencing `JET_CSS_DIRECTORY`:

```
#JET_CSS_DIRECTORY#
```

JET_JS_DIRECTORY

Use the `JET_JS_DIRECTORY` substitution string to reference the JavaScript directory of the Oracle JavaScript Extension Toolkit (JET) components which ships with Oracle Application Express. Supported syntax for referencing `JET_JS_DIRECTORY`:

```
#JET_JS_DIRECTORY#
```

LOGIN_URL

Use `LOGIN_URL` to display a link to a login page for users that are not currently logged in. [Table 2-29](#) describes the supported syntax for `LOGIN_URL`.



See Also:

"[Authentication](#)" and "[Security Attributes Page](#)"

Table 2-29 LOGIN_URL Syntax

| Reference Type | Syntax |
|-----------------------|------------------------------|
| Bind variable | :LOGIN_URL |
| Direct PL/SQL | APEX_APPLICATION.G_LOGIN_URL |
| PL/SQL | V('LOGIN_URL') |
| Substitution string | &LOGIN_URL. |
| Template Substitution | #LOGIN_URL# |

LOGOUT_URL

`LOGOUT_URL` is an application-level attribute used to identify the logout URL. This is a URL that navigates the user to a logout page or optionally directly logs out a user. To create a logout navigation bar entry, add a trailing period to `&LOGOUT_URL` (`&LOGOUT_URL.`). If you are coding a page template, use `#LOGOUT_URL#`. [Table 2-30](#) describes the supported syntax for referencing `LOGOUT_URL`.

Table 2-30 LOGOUT_URL Syntax

| Reference Type | Syntax |
|-----------------------|-----------------|
| Bind variable | :LOGOUT_URL |
| PL/SQL | V('LOGOUT_URL') |
| Substitution string | &LOGOUT_URL. |
| Template substitution | #LOGOUT_URL# |

APP_TEXT\$Message_Name, APP_TEXT\$Message_Name\$Lang

With `APP_TEXT$Message_Name`, `APP_TEXT$Message_Name$Lang` built-in substitution, you can access application defined or system defined text messages, like `APEX_LANG.MESSAGE.Message_Name`. `Message_Name` has to be a valid identifier (A-Z, 0-9, \$, #, _). The optional `$Lang` parameter can be used to access a specific translation, `$Lang` defaults to the current language for the page request.

Table 2-31 APP_TEXT\$Message_Name, APP_TEXT\$Message_Name\$Lang Syntax

| Reference Type | Syntax |
|---------------------|--|
| Bind variable | :APP_TEXT\$Message_Name , :APP_TEXT\$Message_Name\$Lang |
| PL/SQL | V('APP_TEXT\$Message_Name') , V('APP_TEXT\$Message_Name\$Lang') |
| Substitution string | &APP_TEXT\$Message_Name. , &APP_TEXT\$Message_Name\$Lang. |

The following is an example for the default and the French text message `MY_MESSAGE`, using HTML escaping:

```
Default text: &APP_TEXT$MY_MESSAGE!HTML.
Pardon my French: &APP_TEXT$MY_MESSAGE$FR!HTML.
```

PRINTER_FRIENDLY

The value of `PRINTER_FRIENDLY` determines if the Application Express engine is running in print view mode. This setting can be referenced in conditions to eliminate elements not desired in a printed document from a page. [Table 2-32](#) describes the supported syntax for referencing `PRINTER_FRIENDLY`.

Table 2-32 PRINTER_FRIENDLY Syntax

| Reference Type | Syntax |
|---------------------|--|
| Direct PL/SQL | APEX_APPLICATION.G_PRINTER_FRIENDLY (VARCHAR2 DATATYPE) |
| PL/SQL | V('PRINTER_FRIENDLY') |
| Substitution string | &PRINTER_FRIENDLY. |

PROXY_SERVER

`PROXY_SERVER` is an application attribute. The attribute may be used by regions whose source comes from a URL. The following is the correct syntax for a direct PL/SQL reference used when you are writing PL/SQL to access remote web servers from within the database (for example, when using the `utl_http` package shipped with the database).

APEX_APPLICATION.G_PROXY_SERVER

PUBLIC_URL_PREFIX

`PUBLIC_URL_PREFIX` is an application-level attribute that identifies a URL to toggle out of a logged in mode to a public view. [Table 2-33](#) describes the supported syntax for referencing `PUBLIC_URL_PREFIX`.

Table 2-33 PUBLIC_URL_PREFIX Syntax

| Reference Type | Syntax |
|-----------------------|------------------------|
| Bind variable | :PUBLIC_URL_PREFIX |
| PL/SQL | V('PUBLIC_URL_PREFIX') |
| Substitution string | &PUBLIC_URL_PREFIX. |
| Template substitution | #PUBLIC_URL_PREFIX# |

REQUEST

Each application button sets the value of `REQUEST` to the name of the button or to the request value attribute associated with the button, enabling accept processing to reference the name of the button when a user clicks it. In the `f?p` syntax, `REQUEST` may be set using the fourth argument.

`REQUEST` is typically referenced during Accept processing (that is, the processing that occurs when you post a page). [Table 2-34](#) describes the supported syntax for referencing `REQUEST`.

Table 2-34 REQUEST Syntax

| Reference Type | Syntax |
|---------------------|----------------------------|
| Bind variable | :REQUEST |
| Direct PL/SQL | APEX_APPLICATION.G_REQUEST |
| PL/SQL | V('REQUEST') |
| Substitution string | &REQUEST. |

**See Also:**

"Using REQUEST"

Using REQUEST

`REQUEST` is typically referenced during Accept processing (that is, the processing that occurs when you post a page). This section describes additional information about how to use the `REQUEST` substitution string.

- [About Scope and Value of REQUEST for Posted Pages](#)
- [About the When Button Pressed Attribute](#)
- [About Referencing REQUEST Using Declarative Conditions](#)
- [About Using REQUEST for Show Processing](#)
- [About Using BRANCH_TO_PAGE_ACCEPT](#)

**See Also:**

"REQUEST "

About Scope and Value of REQUEST for Posted Pages

When you post a page, you initiate Accept processing. Accept processing consists of computations, validations, processes, and branches. The value of `REQUEST` is available during each phase of the Accept processing. Once an application branches to a different page then `REQUEST` is set to `NULL`.

The value of `REQUEST` is the name of the button the user clicks, or the name of the tab the user selects. For example, suppose you have a button with a name of `CHANGE`, and a label `Apply Change`. When a user clicks the button, the value of `REQUEST` is `CHANGE`.

About the When Button Pressed Attribute

Validations, processes, and branches have a When Button Pressed attribute. This attribute displays as a select list and contains the names of buttons that exist on the current page. If you make a selection from When Button Pressed, you associate the button's `REQUEST` value with the validation, process, or branch.

When you use a button to submit a page, the `REQUEST` value is passed to the page. The Accept processing logic evaluates each validation, process, and branch that uses a When Button Pressed attribute to determine whether the component should run (or fire). When one of these components runs, do not assume that a user actually clicked the associated button and caused the page to be submitted. Keep in mind, that another button using the same request value may have submitted the page. Similarly, JavaScript on the page can also submit the page and pass in a request value.

About Referencing REQUEST Using Declarative Conditions

Many developers reference `REQUEST` using conditions. For example, you may want to reset pagination when a user clicks **Go** on a report page. You can reset pagination by creating an on-submit page process. The page process can be made conditional using the condition `Request = Expression 1`.

To conditionalize an on-submit page process:

1. Under Condition, select the condition type **Request = Expression 1**.
2. In Expression 1, enter **GO**.

About Using REQUEST for Show Processing

You can also use `REQUEST` for Show processing when navigating to a page using `f?p` syntax. For example:

```
f?p=100:1:&APP_SESSION.:GO
```

Remember that the fourth argument in the `f?p` syntax is `REQUEST`. This example goes to application 100, page 1 for the current session, and sets the value of `REQUEST` to `GO`. Any process or region can reference the value of `REQUEST` using Show processing.

The following is a similar example using PL/SQL:

```
IF V ('REQUEST') = 'GO' THEN  
    http.p('hello');  
END IF;
```

Note that `http.p('hello')` is a call to a PL/SQL Web Toolkit package to print the specified text string.



See Also:

Oracle Database Development Guide

About Using BRANCH_TO_PAGE_ACCEPT

You can use a special request `BRANCH_TO_PAGE_ACCEPT` for Show processing to automatically submit the page. For example:

```
f?p=100:1:&APP_SESSION.:BRANCH_TO_PAGE_ACCEPT|SAVE:::P1_DATA:value
```

Using `BRANCH_TO_PAGE_ACCEPT` is the same as navigating to page 1, entering a value into the item `P1_DATA`, and clicking a button that submits the page with a `SAVE` request.

SCHEMA OWNER

If you are generating calls to applications from within your PL/SQL code, you must reference the owner of the Oracle Application Express schema. The following describes the correct syntax for a direct PL/SQL reference:

```
APEX_APPLICATION.G_FLOW_SCHEMA_OWNER
```

You may also use `#FLOW_OWNER#` to reference this value in SQL queries and PL/SQL (for example, in a region or a process).

SQLERRM

`SQLERRM` is a template substitution only available in the Applications Region Error Message. Supported syntax for a region template substitution reference:

```
#SQLERRM#
```

SYSDATE_YYYYMMDD

`SYSDATE_YYYYMMDD` represents the current date on the database server, with the `YYYYMMDD` format mask applied. You may use this value instead of repeated calls to the `SYSDATE()` function. The following list describes the supported syntax for referencing `SYSDATE_YYYYMMDD`.

- Bind variable
`:SYSDATE_YYYYMMDD`
- PL/SQL
`V('SYSDATE_YYYYMMDD')`
- Direct PL/SQL
`APEX_APPLICATION.G_SYSDATE (DATE DATATYPE)`

Table 2-35 SYSDATE_YYYYMMDD Syntax

| Reference Type | Syntax |
|----------------|---|
| Bind variable | <code>:SYSDATE_YYYYMMDD</code> |
| Direct PL/SQL | <code>APEX_APPLICATION.G_SYSDATE (DATE DATATYPE)</code> |
| PL/SQL | <code>V('SYSDATE_YYYYMMDD')</code> |

THEME_DB_IMAGES

Use the `THEME_DB_IMAGES` substitution string to always reference files which are stored with your theme definition in the database, regardless of the File Prefix" setting of your theme. Supported syntax for a template substitution:

```
#THEME_DB_IMAGES#
```

THEME_IMAGES

Use the `THEME_IMAGES` substitution string to reference files which are stored with your theme definition. Supported syntax for a template substitution:

```
#THEME_IMAGES#
```

WORKSPACE_IMAGES

Use this substitution string to reference uploaded images, JavaScript, and cascading style sheets that are shared over many applications within a workspace. [Table 2-36](#) describes the supported syntax for referencing `WORKSPACE_IMAGES`.

Table 2-36 WORKSPACE_IMAGES Syntax

| Reference Type | Syntax |
|----------------|--------------------------------|
| Bind variable | <code>:WORKSPACE_IMAGES</code> |
| Direct PL/SQL | Not available |

Table 2-36 (Cont.) WORKSPACE_IMAGES Syntax

| Reference Type | Syntax |
|-----------------------|-------------------------|
| PL/SQL | V('WORKSPACE_IMAGES') |
| Substitution string | &WORKSPACE_IMAGES. |
| Template substitution | #WORKSPACE_IMAGES# |

**See Also:**

"APP_IMAGES" and "IMAGE_PREFIX"

WORKSPACE_ID

Use this substitution string to reference the workspace ID. [Table 2-37](#) describes the supported syntax for referencing WORKSPACE_ID.

Table 2-37 WORKSPACE_ID Syntax

| Reference Type | Syntax |
|----------------------|---|
| Bind variable | :WORKSPACE_ID |
| PL/SQL | V('WORKSPACE_ID') |
| Substitution string | &WORKSPACE_ID. |
| SYS_CONTEXT variable | SELECT ... WHERE workspace_id = SYS_CONTEXT('APEX\$SESSION', 'WORKSPACE_ID') |

Consider the following examples:

- From within an HTML region:
Hello your workspace id is &WORKSPACE_ID.
- Using PL/SQL:

```
http.p('Hello your workspace id is '||v('WORKSPACE_ID'));
```
- Using a bind variable:

```
SELECT * FROM some_table WHERE workspace_id = :WORKSPACE_ID
```
- Using the SYS_CONTEXT variable:

```
SELECT ... WHERE workspace_id = SYS_CONTEXT('APEX$SESSION', 'WORKSPACE_ID')
```

Oracle Application Express sets up the APEX\$SESSION context when it starts to process an incoming request. For example, you can use the value of 'WORKSPACE_ID' to access the current workspace ID value in queries and VPD (Virtual Private Database) security policies that protect your table data.

3

Utilizing the App Gallery

Oracle Application Express includes a variety of productivity and sample applications. Developers can install and run productivity and sample applications to learn more about the types of applications they can build and how to construct specific types of pages.

- [About App Gallery](#)
The App Gallery includes three types of applications: sample applications, productivity applications, and custom applications.
- [Managing Productivity and Sample Apps in a Full Development Environment](#)
In a full development environment, developers can view, install, run, remove, unlock, export, change authentication, and update productivity and sample apps.
- [Understanding Sample Database Application](#)
Sample Database Application is a sample application that highlights common design concepts and demonstrates the use of reports, charts, calendars, maps, and trees.
- [Understanding AnyCo IT Department Websheet](#)
AnyCo IT Department is a sample application that highlights many Websheet features and capabilities, contains several data grids, and demonstrates how to include data in embedded reports and charts.
- [Utilizing App Administration](#)
Administrators use App Administration to manage and monitor productivity and sample apps in their workspace.
- [Using a Productivity and Sample Apps Only Workspace](#)
A Productivity and Sample Apps Only workspace enables developers to share sample and productivity applications with other developers and administrators.

About App Gallery

The App Gallery includes three types of applications: sample applications, productivity applications, and custom applications.

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

Productivity Applications

Productivity applications are fully functional applications that have been designed to address a specific business need. You can install, run, and use a productivity application as is, or analyze it to better understand how to build a specific type of functionality. Productivity applications are available for installation in a workspace, but are not editable by default.

Sample Applications

As the group name implies, an Oracle-supplied **sample application** is not a complete application but instead contains code snippets or sample code. Sample applications are available for installation in a workspace and are editable by default.

Custom Applications

Custom applications are applications you create and that are managed by your instance administrator. Like Oracle-supplied sample and productivity applications, custom applications typically include both the application pages and supporting database objects. A developer creates a custom application using an application export. Once exported, your instance administrator imports it to make it available. Custom applications are available for installation in a workspace, but are not be editable by default.



See Also:

"Managing Custom Applications" in *Oracle Application Express Administration Guide*

Managing Productivity and Sample Apps in a Full Development Environment

In a full development environment, developers can view, install, run, remove, unlock, export, change authentication, and update productivity and sample apps.

- [Accessing the App Gallery Page](#)
- [Installing a Productivity and Sample App](#)
- [Running a Productivity and Sample App](#)
- [Removing a Productivity and Sample App](#)
- [Changing Authentication for an Installed Productivity and Sample App](#)
- [Unlocking an Installed Productivity Application](#)
- [Updating a Productivity and Sample App](#)



See Also:

"Managing Custom Applications" in *Oracle Application Express Administration Guide* and "[Using a Productivity and Sample Apps Only Workspace](#)"

Accessing the App Gallery Page

 **Tip:**

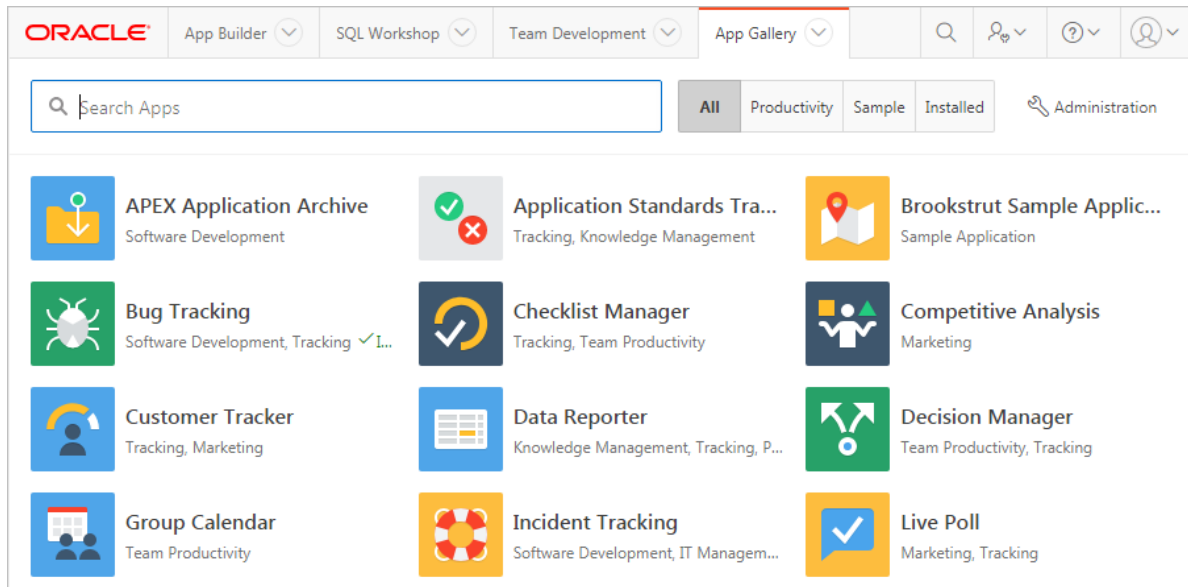
In new workspace, a gray box displays on the **App Builder** home page labeled, **Install a Productivity or Sample App**. Click this box as a shortcut to access the App Gallery.

Use the App Gallery page to view, install, run, remove, unlock, export, change authentication, and update productivity and sample applications.

To access the App Gallery:

1. Sign in to Oracle Application Express.
2. On the Workspace home page, click the **App Gallery** icon.

The App Gallery page appears.



3. Review the available applications.
4. To narrow the display:
 - Click the Search field at the top of the page and enter keywords.
 - Click the following buttons to the right of Search field:
 - **All**. View all applications.
 - **Productivity**. View only productivity applications. Productivity applications are fully functional applications that have been designed to address a specific business need. Once a productivity application is installed, you must unlock it before you can edit it.

- **Sample.** View only sample applications. Sample applications are not a complete application but instead contain code snippets or sample code. By default, sample applications are fully editable.
 - **Installed.** View only installed productivity and sample apps.
5. Select an application.

The App Details page appears, displaying summary information about the application.

On App Details page, you can:

- **Install App.** If not installed, click **Install App** to install it.
- **Manage.** If installed, click **Manage** to change the application authentication, unlock, or remove the application.
- **Run.** If installed, click **Run** to run the application.

 **Tip:**

The options that display on the App Details page depend upon whether the application is installed.

Installing a Productivity and Sample App

 **Tip:**

In a new workspace, a gray box displays on the **App Builder** home page labeled, **Install a Productivity or Sample App**. Click this box as shortcut to access the App Gallery.

To install a Productivity and Sample App:

1. Sign in to Oracle Application Express.
The Workspace home page appears.
2. Click the **App Gallery** icon.
The App Gallery page appears.
3. Locate an application to install. Use the Search field at the top of the page to narrow the display results.
4. Click an application to install.
The App Details page appears.
5. Click **Install App**.
6. Select an Authentication scheme and click **Next**.
7. Click **Install App** again.

The Install Application page appears. The success message, `Application installed`, displays at the top of the page.

8. Click **Manage**.

The Manage dialog displays three buttons:

- **Change Authentication**
- **Unlock Remove**
- **Remove**

 **Tip:**

Unlock is NOT available in a Productivity and Sample Apps Only Workspace.

9. To run the application, click the **Run** icon.

Running a Productivity and Sample App

To run an installed Productivity and Sample App:

1. On the Workspace home page, click the **App Gallery** icon.
The App Gallery page appears.
2. Click the **Installed** tab, select the application, and then click the **Run** icon.
3. Enter the appropriate credentials:
 - a. Username - Enter your workspace username.
 - b. Password - Enter your workspace password.
 - c. Click **Sign In**.

The application displays in a new window.

 **See Also:**

["How Your Browser Impacts the Way Applications Run"](#)

Removing a Productivity and Sample App

To remove an installed Productivity and Sample App:

1. On the Workspace home page, click the **App Gallery** icon.
The App Gallery page appears.
2. Click the **Installed** tab and select the application to be removed.
3. Click **Manage**.
4. Click **Remove**.
5. When prompted, click **Remove Application**.



Tip:

You can also perform this task on the Application Details page. See "[Accessing the App Gallery Page.](#)"

Changing Authentication for an Installed Productivity and Sample App

To change the authentication for an installed productivity and sample app:

1. On the Workspace home page, click the **App Gallery** icon.
The App Gallery page appears.
2. Click the **Installed** tab and select the application.
3. Click **Manage**.
4. Click **Change Authentication**.
The Change Authentication page appears.
5. From Authentication, select an authentication scheme and click **Apply Changes**.



Tip:

You can also perform this task on the Application Details page. See "[Accessing the App Gallery Page.](#)"



See Also:

["Establishing User Identity Through Authentication"](#)

Unlocking an Installed Productivity Application

The main difference between an Oracle-supplied *sample* and *productivity* application is the level of support. Once a productivity application is installed, you must unlock it before you can edit it. By default, sample applications are fully editable.



Note:

Unlocking a productivity application makes it ineligible for future upgrades or support by Oracle Support.

To unlock an installed productivity application:

1. On the Workspace home page, click the **App Gallery** icon.

The App Gallery appears.

2. Click the **Installed** tab and select the application.
3. Click **Manage**.
4. Click **Unlock**.

 **Tip:**

Unlock is NOT available in a Productivity and Sample Apps Only Workspace.

The Unlock Application page appears.

5. Click **Unlock Application**.

 **See Also:**

["Installing a Productivity and Sample App"](#)

Updating a Productivity and Sample App

You can update a productivity and sample app if an Update button displays on the App Details page.

To update a productivity and sample app:

1. On the Workspace home page, click the **App Gallery** icon.

The App Gallery page appears.

2. Click the **Installed** tab and select the application.
3. Click **Update Application**.

The Update Application page appears.

 **Note:**

Because sample applications are unlocked by default, they cannot be updated. To access the latest sample applications, Oracle recommends deleting the sample application from your workspace and reinstalling from the App Gallery.

 **See Also:**

["Running a Productivity and Sample App"](#)

Understanding Sample Database Application

Sample Database Application is a sample application that highlights common design concepts and demonstrates the use of reports, charts, calendars, maps, and trees.

Tip:

Demonstration applications and demonstration database objects are not created by default when creating a new workspace. To change this behavior, an instance administrator must change Feature Configuration settings in the Administration Services application. See "Enabling the Creation of Demonstration Objects in New Workspaces" and "Enabling the Creation of Websheet Objects in New Workspaces" in *Oracle Application Express Administration Guide*

- [Installing and Running Sample Database Application](#)
Install sample applications such as *Sample Database Application* on the App Gallery page.
- [Reviewing Sample Database Application](#)
Sample Database Application demonstrates how to display summary information, use reports and forms for viewing, updating, and adding information, include charts and maps, and create dedicated mobile pages that replicate most functionality available in the desktop version.
- [Modifying Sample Database Application](#)
Learn how to build applications by studying sample applications such as *Sample Database Application*.

Installing and Running Sample Database Application

Install sample applications such as *Sample Database Application* on the App Gallery page.

To install *Sample Database Application*:

1. Sign in to Oracle Application Express.
The Workspace home page appears.
2. Click the **App Gallery** icon.
The App Gallery page appears.
3. Locate and select *Sample Database Application*.

Tip:

Use the Search field at the top of the page to narrow the display.

The App Details page appears.

4. Click **Install App**.

5. Select an Authentication scheme and click **Next**.
6. Click **Install App** again.
The Install Application page appears. The success message, *Application installed*, displays at the top of the page.
7. To run the application, click **Run**. Enter the appropriate login credentials:
 - a. Username - Enter your workspace username.
 - b. Password - Enter your workspace password.
 - c. Click **Sign In**.



Tip:

Once you install a sample application, it displays under Installed on the App Gallery page.

Reviewing Sample Database Application

Sample Database Application demonstrates how to display summary information, use reports and forms for viewing, updating, and adding information, include charts and maps, and create dedicated mobile pages that replicate most functionality available in the desktop version.

- [Home Page](#)
- [About the Navigation Menu](#)
- [Customers Page](#)
- [Using the Customers Page](#)
- [Products Page](#)
- [Using the Products Page](#)
- [Orders Page](#)
- [Using the Orders Page](#)
- [Reports Page](#)
- [Administration Page](#)

Home Page

The screenshot shows the home page of the 'Sample Database Application'. The interface includes a navigation sidebar on the left with links to Home, Customers (7), Products (10), Orders (10), Reports, and Administration. The main content area features a search bar and a dashboard with four key metrics: Monthly Sales (\$0), Monthly Orders (0), Total Products (10), and Total Customers (7). Below the dashboard are two interactive reports: 'Top Customers' and 'Top Products'. The 'Top Customers' report shows 'Bradley, Eugene' with a value of 2,760. The 'Top Products' report shows 'Jacket - 18 x \$150' for \$2,700 and 'Bag - 16 x \$125' for \$2,000.

The home page contains the following main regions:

- **Dashboard** - Demonstrates the use of a Badge List plug-in. This region displays a value based on an underlying SQL statement. Links in this region take you to reports on Monthly Sales, Monthly Orders, Total Products, and Total Customers.
- **Top Customers** - Demonstrates the Slider Tooltip plug-in. This region displays a value based on an underlying SQL statement. Links in this region take you to reports on Monthly Sales, Monthly Orders, Total Products, and Total Customers.
- **Top Products** - Top Products is an interactive report. The report is based on an underlying SQL statement and displays a subset of the information that appears on the Customers page. Users can view the additional details by clicking the product or by clicking the view Products icon (a right arrow) in the upper right corner of the region.
- **Top Orders by Date** - Demonstrates the use of a HTML5 Bar Chart plug-in. This chart displays orders by date and order amount, based on an underlying SQL statement. Users can view the Orders page by clicking the right arrow in the upper right corner.
- **Tags** - Demonstrates the use of a Tag Cloud plug-in. This region displays a value based on an underlying SQL statement. Links in this region take you to search results matching the selected tag.

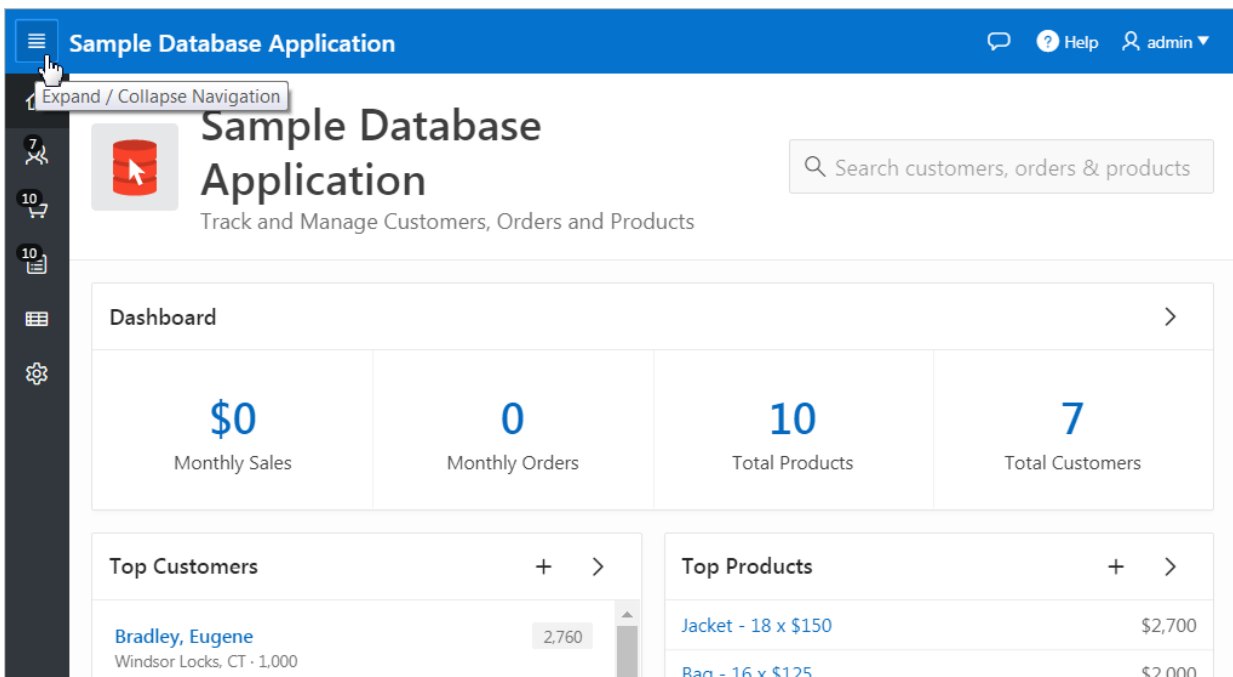
 See Also:

- ["Developing Reports"](#)
- ["Managing Regions "](#)
- ["Creating Lists"](#)

About the Navigation Menu

Applications using newer themes, such as Universal Theme - 42, provides navigation with a navigation menu. *Sample Database Application* includes a navigation menu on the left side of the page. Use this menu to move between application pages.

A **Expand and Collapse Navigation** icon displays to the left of the application title and functions as a toggle to collapse and expand the navigation menu. The following illustration shows that the Home page with the navigation menu collapsed.



The screenshot shows the 'Sample Database Application' dashboard. At the top left, there is a blue header with the application title and a user profile 'admin'. Below the header, a dark vertical navigation bar is shown in its collapsed state. A mouse cursor is hovering over the 'Expand / Collapse Navigation' icon (a hamburger menu icon) at the top of this bar. The main content area features a search bar, a dashboard with four summary cards (Monthly Sales: \$0, Monthly Orders: 0, Total Products: 10, Total Customers: 7), and two expandable sections for 'Top Customers' and 'Top Products'.

| Monthly Sales | Monthly Orders | Total Products | Total Customers |
|---------------|----------------|----------------|-----------------|
| \$0 | 0 | 10 | 7 |

| Top Customers | Top Products |
|---|-----------------------------|
| Bradley, Eugene Windsor Locks, CT · 1,000 | Jacket - 18 x \$150 \$2,700 |
| 2,760 | Baq - 16 x \$125 \$2,000 |

To expand the navigation menu again, click the icon again.

 See Also:

- ["Managing Navigation Menus"](#)

Customers Page

The Customers page enables users to view and edit customer information.

| Customer Name | Address | City |
|---------------------|-----------------------------------|-------------|
| Dulles, John | 45020 Aviation Drive | Sterling |
| Hartsfield, William | 6000 North Terminal Parkway | Atlanta |
| Logan, Edward | 1 Harborside Drive | East Boston |
| OHare, Frank | 10000 West OHare | Chicago |
| LaGuardia, Fiorello | Hangar Center, Third Floor | Flushing |
| Lambert, Albert | 10701 Lambert International Blvd. | St. Louis |
| Bradley, Eugene | Schoephoester Road | Windsor Loc |

Customers is an interactive grid tracking customer information. To search for a customer information, enter search terms in the Search field and click **Go**. To sort by customer, click the column heading and then select the sort icons.

You can change the appearance of the report using the Actions menu.



See Also:

["Managing Interactive Grids"](#)

Using the Customers Page

Use the Customers page to edit existing customers, create new customers, or upload data.

- [Editing a Customer](#)
- [Creating a Customer](#)
- [Uploading Data](#)

Editing a Customer

To edit an existing customer:

1. Click **Customers** in the navigation menu.
The Customers page appears
2. To update a customer:
 - a. Click the customer name.
 - b. On the Customer Details page, edit the fields and click **Apply Changes**.
3. To delete a customer:
 - a. Click the customer name.
 - b. On the Customer Details page, click **Delete**.

Creating a Customer

To add a new customer:

1. Click **Customers** in the navigation menu.
The Customers page appears
2. Click the **Create Customer** button.
The Customer Details page appears.
3. Add the appropriate details and click **Add Customer**.

Uploading Data

To upload data:

1. Click **Customers** in the navigation menu.
The Customers page appears
2. Click **Upload Data**.
3. For Import From, select either:
 - **Upload file, comma separated (*.csv) or tab delimited.**
 - **Copy and Paste.**The UI changes based on the import option you select.
4. If you select **Upload file, comma separated (*.csv) or tab delimited**:
 - a. File Name - Click and locate the file to upload.
 - b. Separator - Identify a column separator character. Use \t for tab separators.
 - c. Optionally Enclosed By - Enter a delimiter character. See field-level Help for more details.
 - d. First Row has Column Names - Select or deselect **Yes**.
 - e. Use Advanced Settings - Select **Yes** to view the additional settings.
 - f. Click **Next**.
5. If you select **Copy and Paste**:
 - a. Copy and Paste Delimited Data- Copy and paste into the field provided.
 - b. Separator - Identify a column separator character. Use \t for tab separators.

- c. Optionally Enclosed By - Enter a delimiter character. See field-level Help for more details.
 - d. First Row has Column Names - Select or deselect **Yes**.
 - e. Use Advanced Settings - Select **Yes** to view the additional settings.
 - f. Click **Next**.
6. On Data/Table Mapping, review the mappings and click **Next**.
 7. On Data Validation, review the data and click **Load Data**.
 8. Review the results.
 9. Click **Finish**.

Products Page

Products page enables users to view and edit product information. The Products page features an interactive report based on a SQL query that uses a custom function for displaying images stored in the database.

| Image | Name ↑ | Category | Available | Price | Units | Sales | Customers |
|-------|--------|-------------|-----------|----------|-------|------------|-----------|
| | Bag | Accessories | Yes | \$125.00 | 16 | \$2,000.00 | 6 |
| | Belt | Accessories | Yes | \$30.00 | 11 | \$330.00 | 3 |
| | Blouse | Womens | Yes | \$60.00 | 16 | \$960.00 | 5 |

By default, this page displays in Report view. You can change the view by clicking **View Icons** and **View Detail** to the left of the Actions menu. In Report view, you can sort by column by clicking the column heading and then selecting the icons. Users can change the appearance of the report using the Actions menu.



See Also:

"Customizing Interactive Reports in a Running Application" and "About the Actions Menu"

Using the Products Page

Use the Products page to edit or add products.

- [Editing a Product](#)
- [Adding a Product](#)

Editing a Product

To edit or delete a product:

1. Click **Products** in the navigation menu.
2. To update a product:
 - a. On the Products page, click the product name.
 - b. On the Product Details page, edit the fields and click **Apply Changes**.
3. To delete a product:
 - a. On the Products page, click the product name.
 - b. On the Product Details page, click **Delete**.

Adding a Product

To add a new product:

1. Click **Products** in the navigation menu.
2. Click the **Add Product** button.
3. On the Product Details page, edit the fields and click **Add Product**.

Orders Page

The Orders page enables users to create, view, and edit customer orders. The Orders page is an interactive report for tracking order information. To sort by column, click the column heading and then select the appropriate sort icon. You can change the appearance of the report using the Actions menu. To edit an order, click the Order #.

| Order # | Customer Name | Order Date | Order Items | Order Total | Sales Rep | Tags |
|---------|---------------------|------------|-------------|-------------|-----------|------|
| 0010 | Bradley, Eugene | 3/26/2018 | 3 | \$870.00 | DEMO | - |
| 0009 | Hartsfield, William | 3/23/2018 | 3 | \$730.00 | DEMO | - |
| 0008 | OHare, Frank | 3/17/2018 | 4 | \$1,060.00 | DEMO | - |
| 0007 | Logan, Edward | 3/9/2018 | 7 | \$905.00 | DEMO | - |
| 0006 | Logan, Edward | 3/4/2018 | 4 | \$1,515.00 | DEMO | - |



See Also:

["Managing Interactive Reports"](#)

Using the Orders Page

Use the Orders page to edit or place orders.

- [Editing an Order](#)
- [Placing an Order](#)

Editing an Order

To edit or delete an order:

1. Click **Orders** in the navigation menu.
2. To update an order:
 - a. Click the Order number.
The Order Details page appears.
 - b. Edit the cells and click **Apply Changes**.
3. To delete an order:
 - a. Click the Order number.
The Order Details page appears.
 - b. Click the check box next to product to be deleted and click **Delete**.

Placing an Order

To create an order:

1. Click **Orders** in the navigation menu.
2. Click the **Place order** button.
The Identify Customer wizard appears.
3. Follow the on-screen instructions to enter order information.
4. Click **Complete Order**.
The order confirmation page appears.
5. Click **Close**.

Reports Page

Access the Reports page to view information in various formats, including bar chart, pie chart, map, tree, and tags. To view a report or chart, select it.

The screenshot shows the 'Reports' page of the 'Sample Database Application'. The top navigation bar includes 'Sample Database Application', 'Help', and 'admin'. The left sidebar menu has 'Reports' selected, showing sub-items: Sales by Category, Sales by Month, Sales by Product, Sales by State, Product Order Tree, Customer Orders, and Tags. The main content area, titled 'Reports', contains eight report cards:

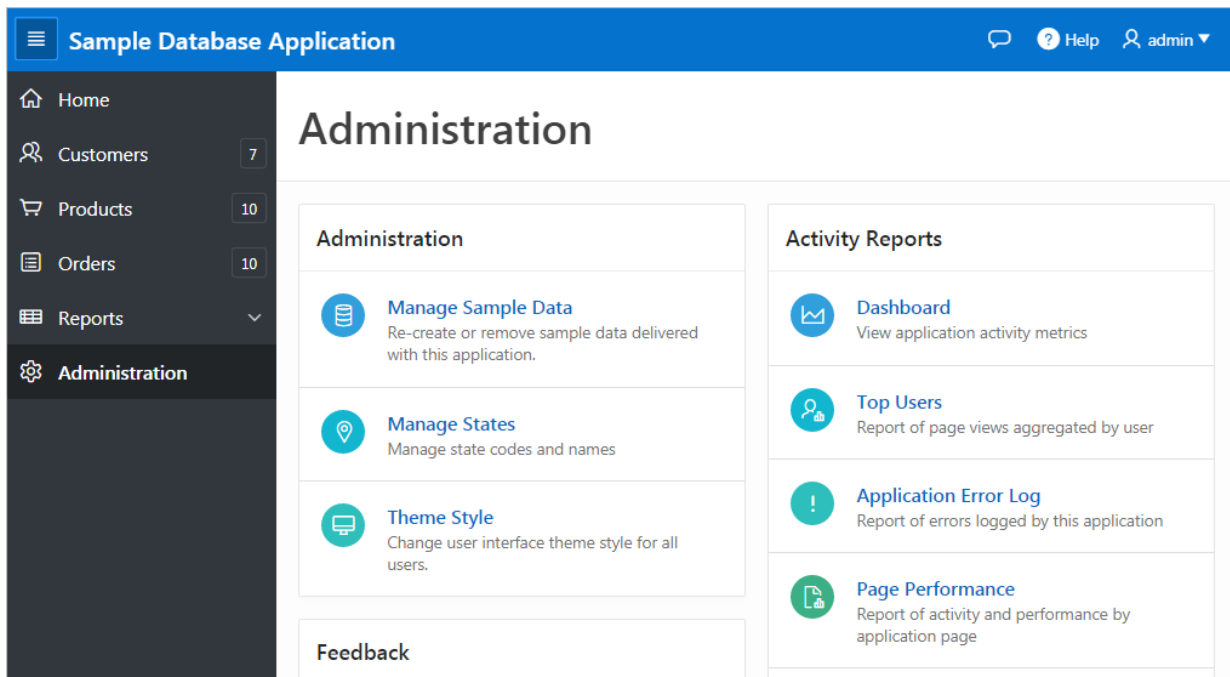
- Sales by Category**: Pie chart of sales for each category.
- Customer Map**: Map of orders with zoom in and out.
- Sales by Product**: Bar chart of sales for each product.
- Product Order Tree**: Tree control with expand and collapse.
- Sales by Month**: Vertical bar chart of sales by month.
- Customer Orders**: Stacked bar chart of customer orders by Month.
- Sales by State**: CSS chart of sales by State.
- Tags**: View tags by customer, product, and order.

See Also:

- ["Creating Charts"](#)
- ["Creating Maps"](#)
- ["Managing Trees"](#)

Administration Page

Use the Administration page to manage sample data, manage state codes and names, edit the defined Theme Style, and enable and manage user feedback, and view activity and performance reports.



The Administration page features the following regions:

- **Manage Sample Data** - Re-create or remove sample data delivered with this application.
- **Manage States** - Manage state codes and names.
- **Theme Styles** - Change user interface theme styles for all users.
- **Feedback Settings** - Manage if attachment should be allowed.
- **User Feedback** - View a report of all submitted feedback.
- **Top Users** - Displays top views aggregated by user.
- **Application Error Log** - Report of errors logged by this application.
- **Page Performance** - Reports of activity and performance by application page.
- **Page Views** - Report of each page view by user including date of access and elapsed time.

 **See Also:**

- ["Managing Feedback"](#)
- ["Using Theme Styles and Theme Roller"](#)

Modifying Sample Database Application

Learn how to build applications by studying sample applications such as *Sample Database Application*.

- [Why Modify a Sample Database Application?](#)
- [Editing Sample Database Application](#)

 **See Also:**

["Managing Productivity and Sample Apps in a Full Development Environment"](#)

Why Modify a Sample Database Application?

Once you understand the type of functionality available in a sample application, such as the *Sample Database Application*, the next step is to learn more about the construction of each page. An efficient way to speed up the learning process is to analyze and deconstruct the pages in the applications. If you happen to break something, you can quickly delete the application and install it again.

 **Tip:**

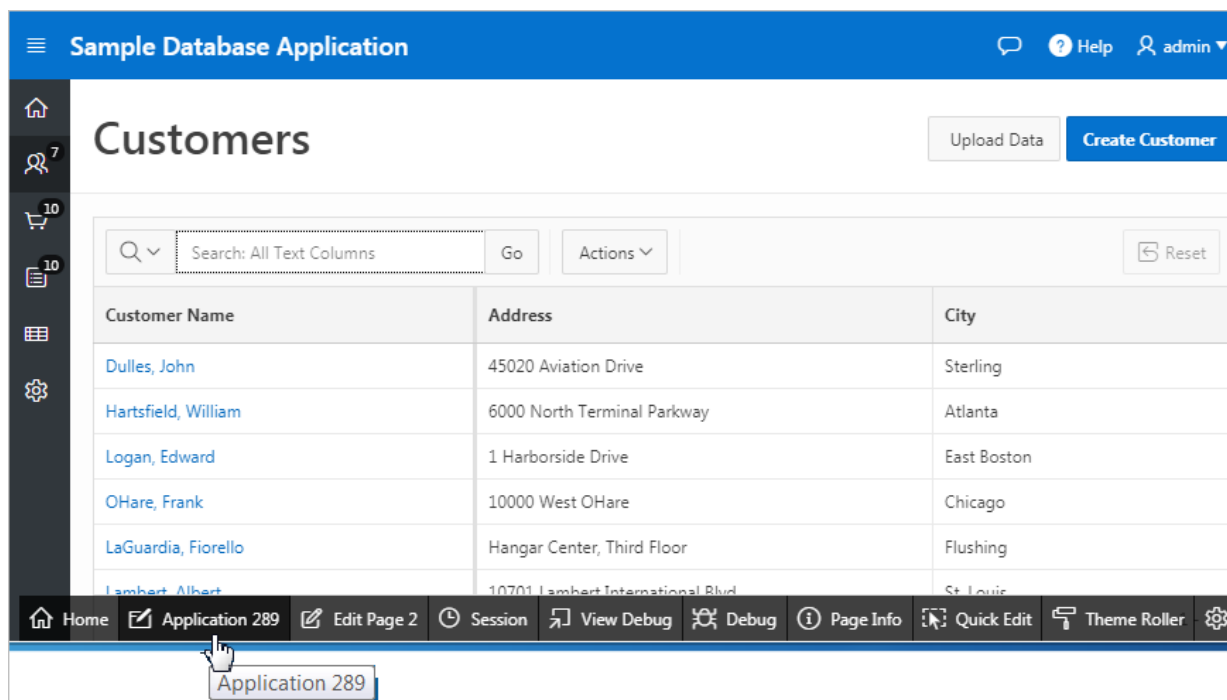
The App Gallery includes *sample* and *productivity* applications. By default, sample applications are fully editable. In contrast, you must unlock a productivity application before you can edit it. Unlocking a productivity application makes it ineligible for future upgrades or support by Oracle Support.

Editing Sample Database Application

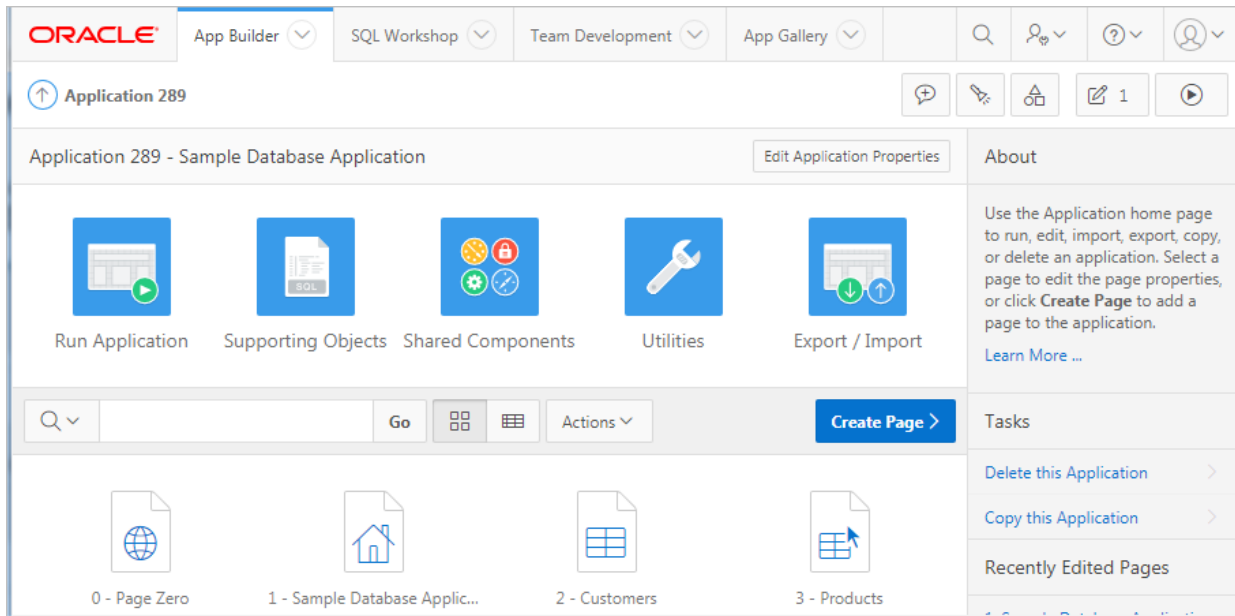
 **Tip:**

When you create a desktop application page, you choose a Page Mode: Normal or Modal. A **Normal** page is a standard Oracle Application Express page. A **Modal dialog** is an overlay window which remains active until the end user closes it. When you run an editable Normal page, the Runtime Developer toolbar displays at the bottom of the page. Use the Developer toolbar to quickly edit the current application or currently running Normal page, or view session state.

To edit a running application, click the **Application** link on the Runtime Developer toolbar.



The Application home page appears. The application ID and application name display at the top of the page.



Use the following icons in the center of the page:

- **Run Application** - Submits current application pages to the Application Express engine to render viewable HTML. See "[How the Application Express Engine Renders and Processes Pages.](#)"
- **Supporting Objects** - Links to the Supporting Objects page. See "[How to Create a Custom Application.](#)"
- **Shared Components** - Links to a list of shared components and user interface controls that can display or be applied on every page within an application. See "[Managing Shared Components.](#)"
- **Utilities** - Links to the Utilities page. Use this page to monitor developer activity, view dashboards, run Advisor, and view numerous other reports. See "[Using Application Utilities](#)"
- **Export/Import** - Links the Export/Import Wizard. Use this wizard to import and export an entire application and related files such as cascading style sheets, images, static files, script files, themes, user interface defaults, and workspace users. See "[Exporting an Application and Application Components.](#)"

Export/Import

The pages that comprise the application display on the bottom of Application home page. To access a specific page, simply click it. To search for a specific page, enter a case insensitive query for the page title or page number in the Search field and click **Go**.

See Also:

"[Understanding the Application Home Page](#)" and "[Runtime Developer Toolbar](#)"

Understanding AnyCo IT Department Websheet

AnyCo IT Department is a sample application that highlights many Websheet features and capabilities, contains several data grids, and demonstrates how to include data in embedded reports and charts.

Like all sample applications, *AnyCo IT Department* is fully editable.

- [Installing and Running AnyCo IT Department](#)
Install sample Websheets such as *Sample Websheet - AnyCo IT Department* on the App Gallery page.
- [Understanding Websheet Navigation](#)
Navigate through Websheet pages by expanding and collapsing sections and clicking breadcrumbs.
- [Understanding AnyCo IT Department](#)
Sample Websheet - AnyCo IT Department demonstrates how users can share text, data, and images using Websheet pages.
- [Modifying Websheet Content](#)
Learn how to build a Websheet applications by studying a sample Websheet such as *AnyCo IT Department*.



See Also:

"[Creating Websheet Applications](#)" and "Using Websheets" in *Oracle Application Express End User Guide*

Installing and Running AnyCo IT Department

Install sample Websheets such as *Sample Websheet - AnyCo IT Department* on the App Gallery page.

To install and run *Sample Websheet - AnyCo IT Department*:

1. Sign in to Oracle Application Express.
The Workspace home page appears.
2. Click the **App Gallery** icon.
The App Gallery page appears.
3. Locate and select the *Sample Websheet - AnyCo IT Department* application.

Tip:

Use the Search field at the top of the page to narrow the display.

The App Details page appears.

4. Click **Install App**.

5. Select an Authentication scheme and click **Next**.
6. Click **Install App**.
A success message appears.
7. To run the application, click the **Run** icon.
8. Enter the appropriate login credentials:
 - Username - Enter your workspace username.
 - Password - Enter your workspace password.
 - Click **Sign In**.The home page displays in a new window.



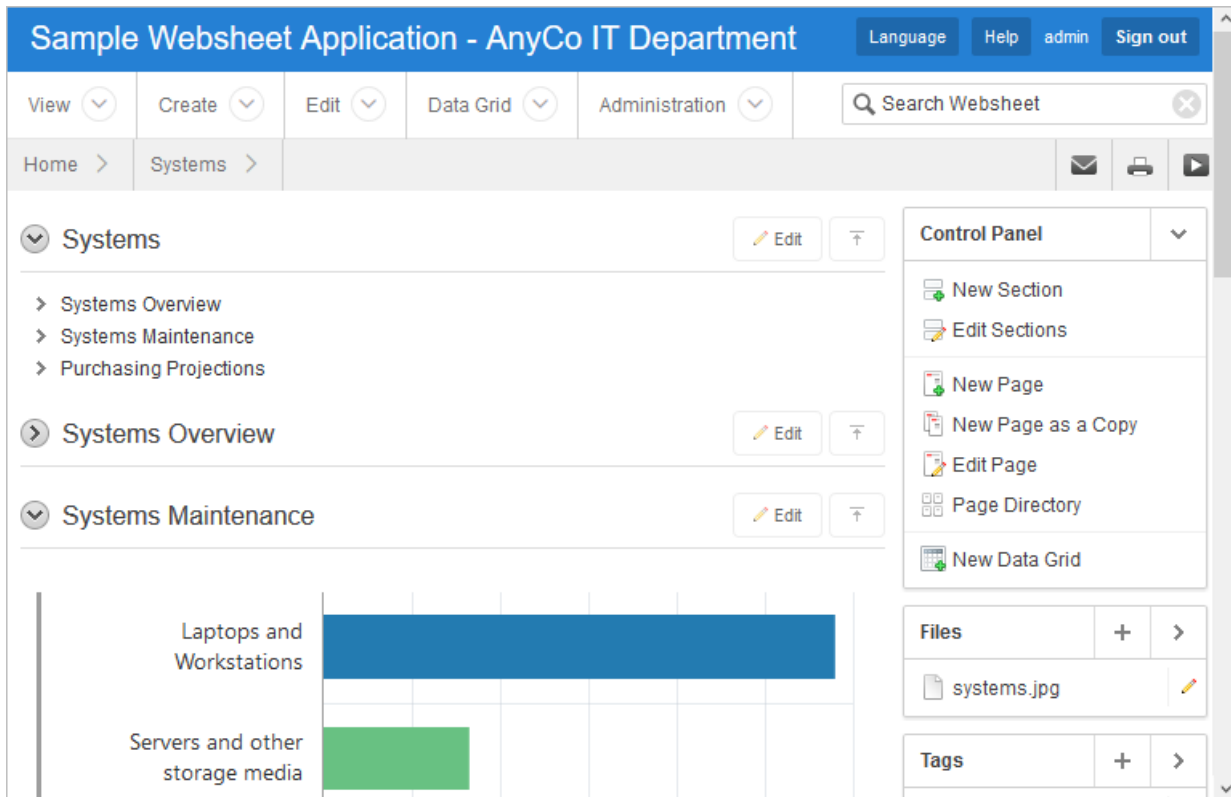
See Also:

["How Your Browser Impacts the Way Applications Run"](#)

Understanding Websheet Navigation

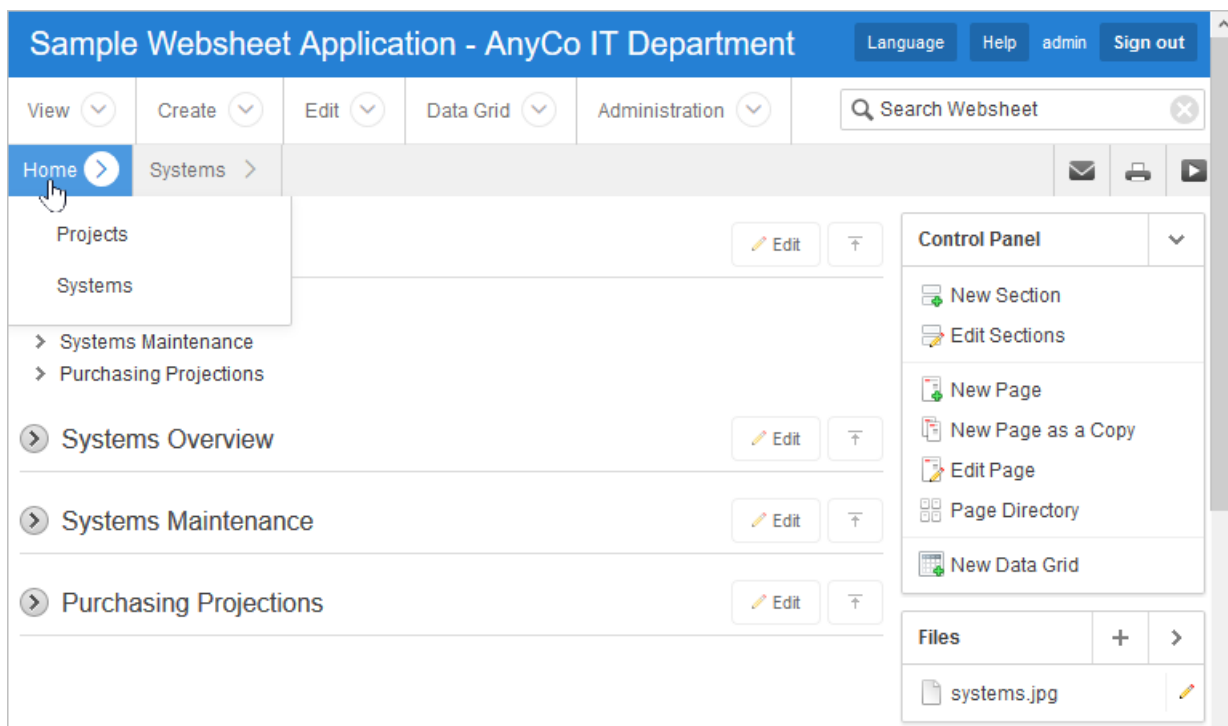
Navigate through Websheet pages by expanding and collapsing sections and clicking breadcrumbs.

Every Websheet contains common navigation elements that enable you to quickly move within a page or between pages. Each Websheet page is divided into sections. You can expand and collapse a page section by clicking the arrow to the left of the Section title. In the following example, Systems Overview has been collapsed.



Using Breadcrumbs to Navigate Between Websheet Pages

A breadcrumb displays at the top of every Websheet page. Clicking the **Home** breadcrumb displays a menu of links to all the top level pages in the Websheet.



Understanding AnyCo IT Department

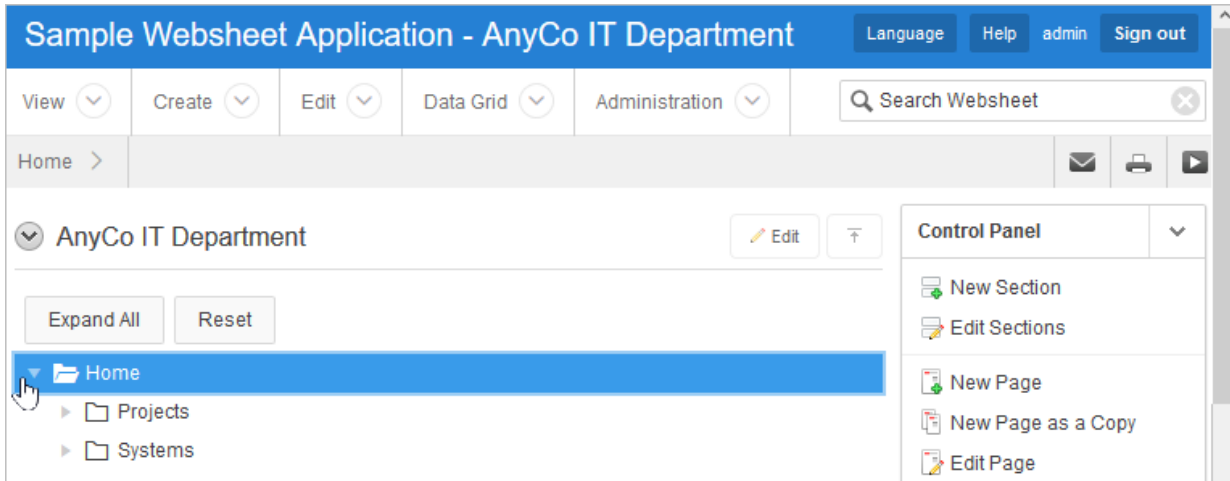
Sample Websheet - AnyCo IT Department demonstrates how users can share text, data, and images using Websheet pages.

This Websheet includes hierarchical navigation between page sections and pages, formatted text that includes easily formatted links, Data Grids that display of tabular data, and content displayed in reports, interactive reports, and charts.

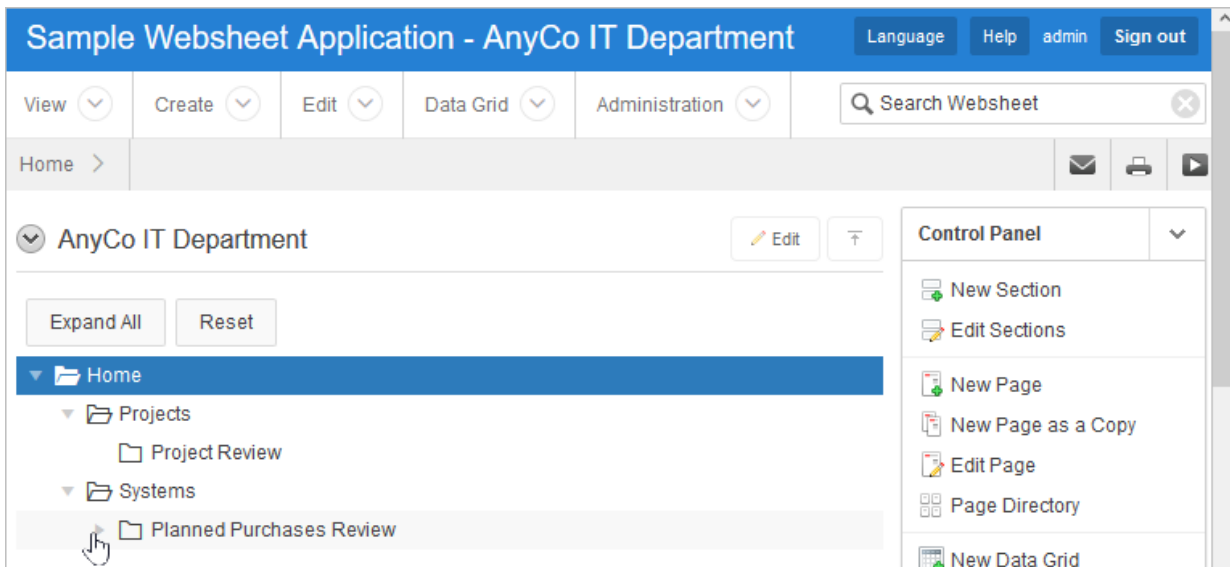
- [Navigating Between Pages and Viewing Page Sections](#)
- [Home Page](#)
- [Projects Page](#)
- [Project Review Page](#)
- [Systems Page](#)
- [Planned Purchase Review](#)
- [Systems Budget Review](#)

Navigating Between Pages and Viewing Page Sections

The hierarchical tree displays at the top of the home page and enables you to navigate between Websheet pages. Click the arrow to the left of the Home breadcrumb to expand or collapse the tree.



To view subsections (or children) within a section, click the arrow to the left of a section name. Click **Expand All** to view all pages and page subsections and **Reset** to return the tree to the default display. To go to another page, select the page name.



Home Page

In addition to the navigation tree at the top of the page, the *Sample Websheet - AnyCo IT Department* home page contains three sections: AnyCo IT Department, Projects Summary, and Systems Summary.

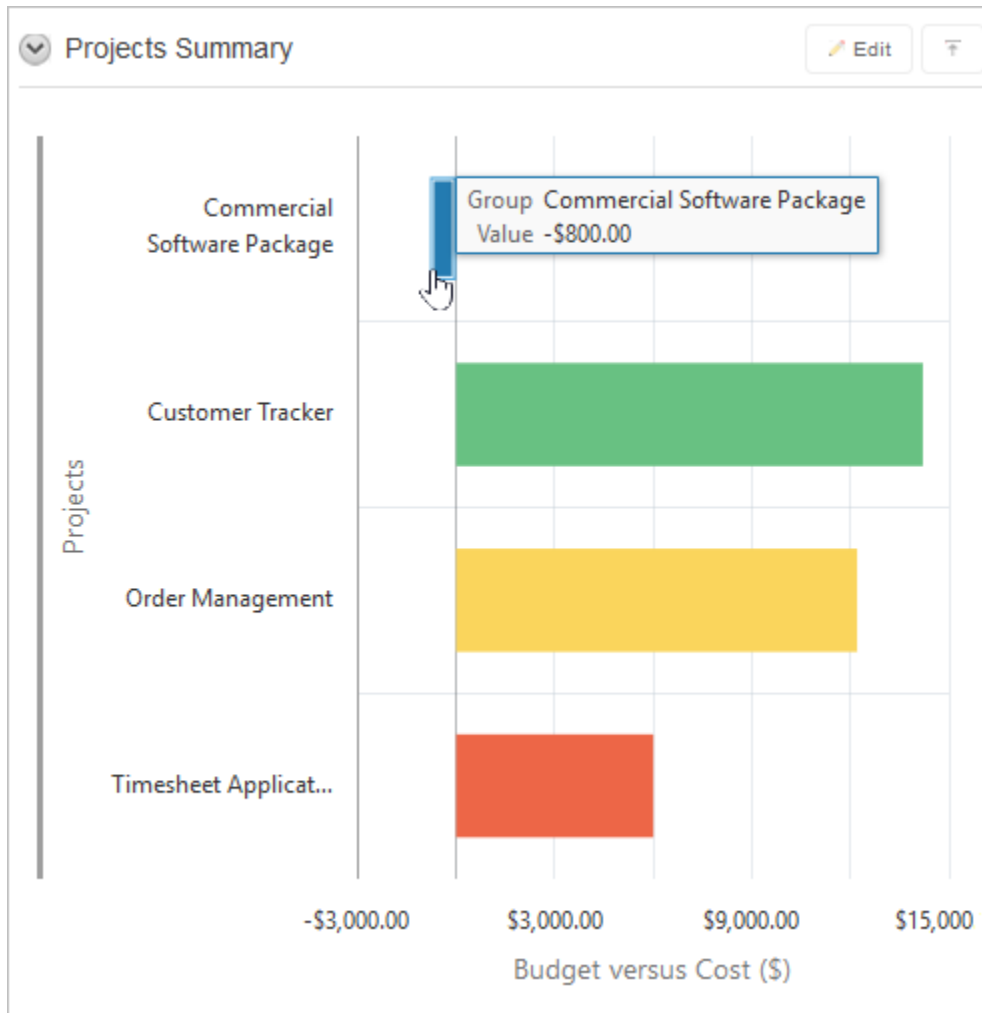
AnyCo IT Department

AnyCo IT Department is a text section that includes links to other pages and an embedded image.

The screenshot shows the 'Sample Websheet Application - AnyCo IT Department' interface. At the top, there is a blue header with the title and navigation links for 'Language', 'Help', 'admin', and 'Sign out'. Below the header is a toolbar with buttons for 'View', 'Create', 'Edit', 'Data Grid', and 'Administration', along with a search box labeled 'Search Websheet'. The main content area is divided into several sections. On the left, there is a breadcrumb trail 'Home >' and a section titled 'AnyCo IT Department' with an 'Edit' button and a filter icon. Below this, there are 'Expand All' and 'Reset' buttons, followed by a tree view showing 'Home' expanded to reveal 'Projects' and 'Systems' sub-items. Another 'AnyCo IT Department' section with an 'Edit' button and filter icon is visible below. The main text area contains two paragraphs: 'AnyCo Corp is a consulting firm founded in 2008. AnyCo Corp specializes in delivering custom solutions to clients in the services industry. With a total of 125 employees, AnyCo Corp is expanding its business to extend consulting services and solutions to clients across North America.' and 'This websheet summarizes the four major projects the IT team is focusing on and the systems required to run the department.' On the right side, there is a 'Control Panel' with options like 'New Section', 'Edit Sections', 'New Page', 'New Page as a Copy', 'Edit Page', 'Page Directory', and 'New Data Grid'. Below the control panel is a 'Files' section showing 'Resources.xlsx' and 'logo.gif' with edit icons.

Projects Summary

Projects Summary is a data grid that displays a chart. Passing the cursor over a bar changes the color and displays a dollar amount.



Systems Summary

Systems Summary is a report of hardware, software, and cloud services. The bottom of the section includes links to Systems Management report and to the Systems page.

▼ Systems Summary Edit Filter

Systems include hardware, software and cloud services. The IT team has classified 6 items. Below is a summary of the Total Costs:

| System | Name | Total Cost Incurred |
|----------|---------------------------------|---------------------|
| Cloud | InitCloud | \$4,300.00 |
| Hardware | Servers and other storage media | \$33,000.00 |
| Hardware | Laptops and Workstations | \$115,800.00 |
| Hardware | Printers, VoIP and Telephony | \$10,600.00 |
| Software | Database, and modeling tool | \$18,400.00 |
| Software | OS and desktop tools | \$12,350.00 |

Projects Page

The Project contains one section, Projects Breakdown. The Projects Breakdown section includes a link to the Project Review page and an updatable report of major IT projects. To edit a specific project, click the **Edit Row** icon. To add another project, click the **Add Row** button.

Sample Websheet Application - AnyCo IT Department Language Help admin Sign out



View ▼ Create ▼ Edit ▼ Data Grid ▼ Administration ▼ Search

Home > Projects >

▼ Projects Breakdown Edit Filter

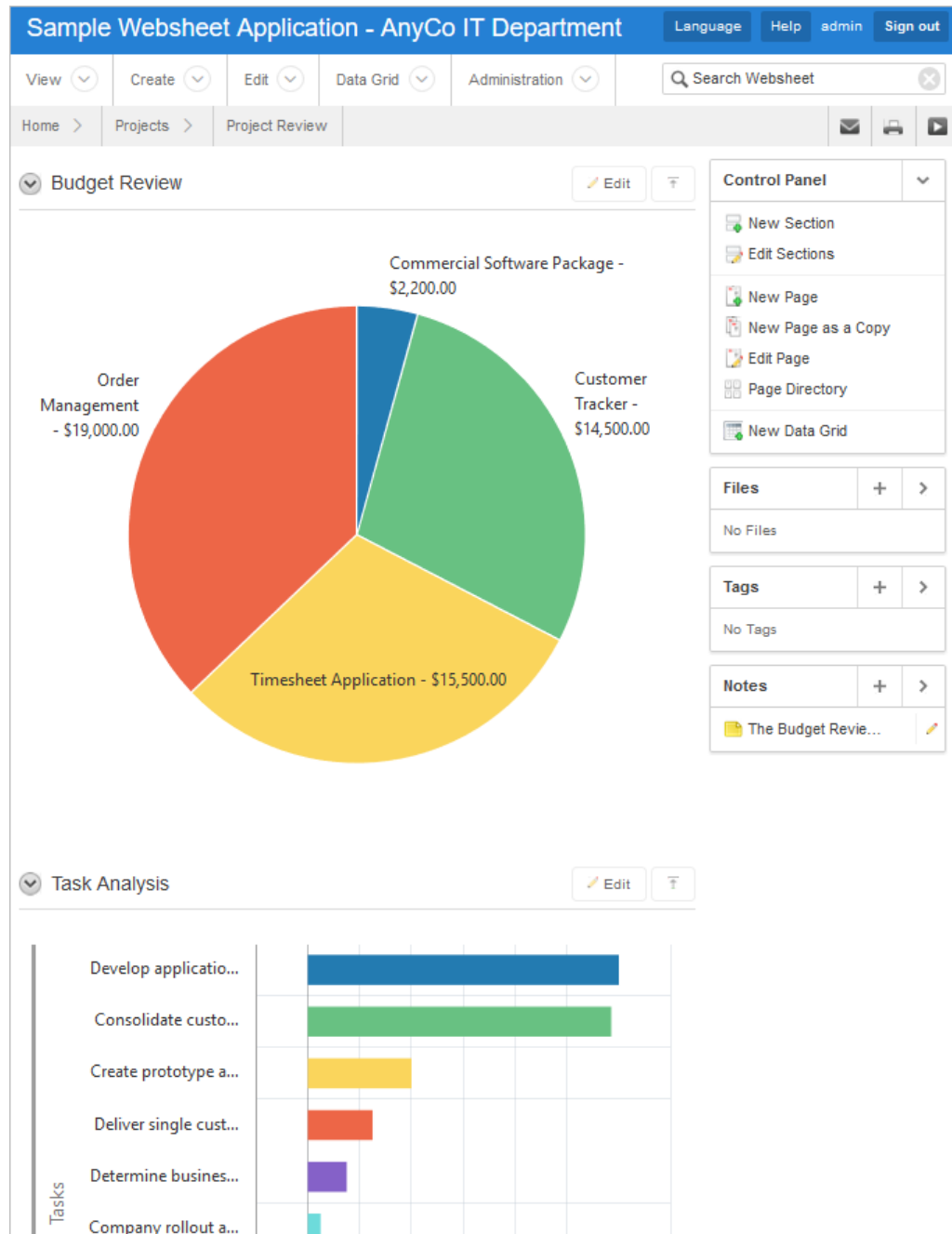
Below are the major projects the IT department are involved with. Also see project summaries within Project Review .

Search Add Row

| | Project | Task | Start Date | End Date | Status | Assigned To | Cost | Budget |
|---|-----------------------------|----------------------------------|-------------|-------------|--------|-------------|------|--------|
|  | Timesheet Application | Company rollout and training | 25-MAR-2010 | 05-DEC-2010 | Open | Pam King | 1000 | 1500 |
|  | Commercial Software Package | Install and customize parameters | 07-APR-2010 | 07-APR-2010 | closed | John Watson | 1000 | 700 |

Project Review Page

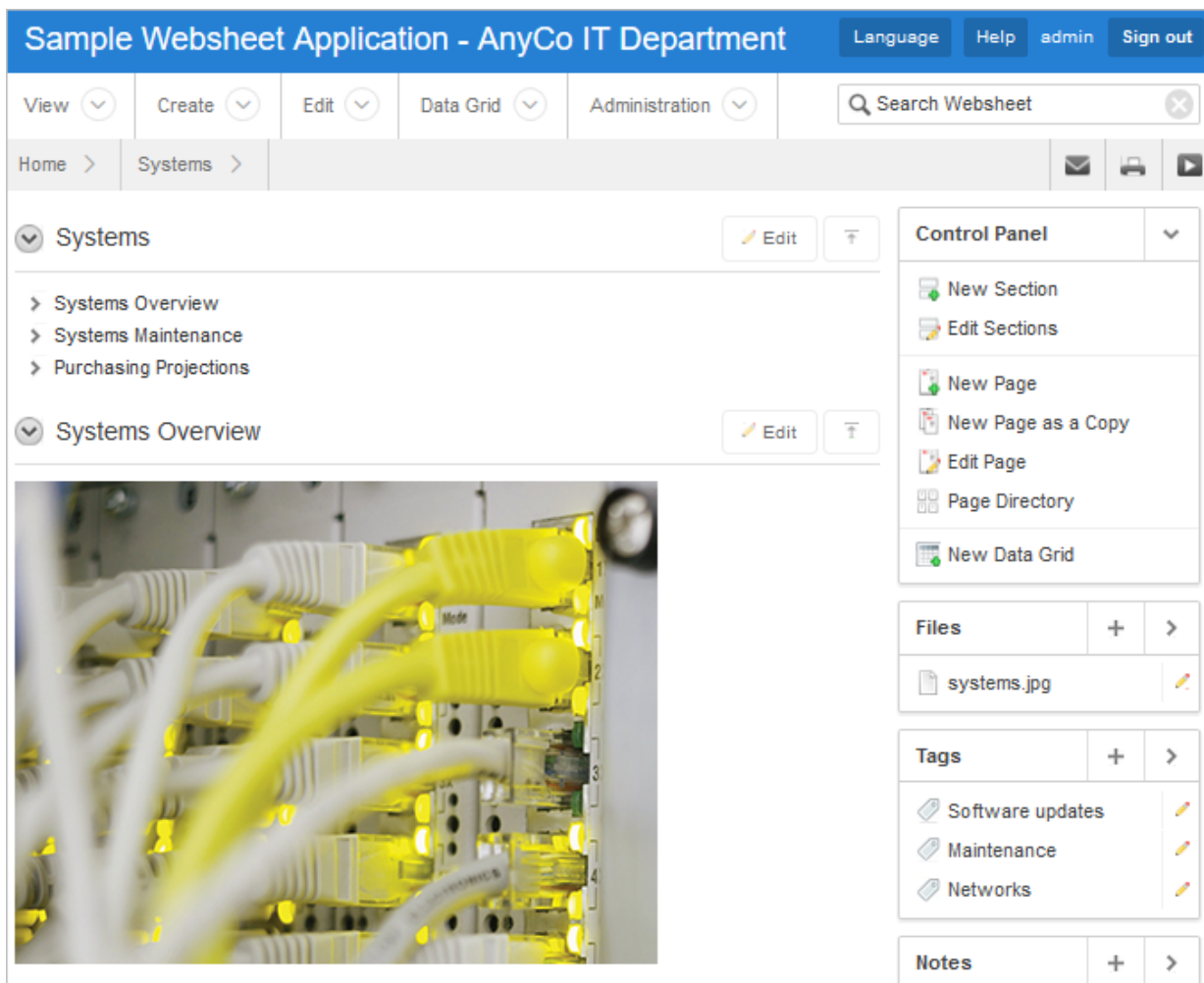
The Project Review page contains two sections, **Budget Review** and **Task Analysis**. Both sections are data grids that display as different types charts.



Systems Page

The Systems page contains the following sections:

- **Systems** is a navigation section that includes links to the sections, Systems Overview, Systems Maintenance, and Purchasing Projections.
- **Systems Overview** is a text section that includes an embedded image.
- **Systems Maintenance** is a data grid that displays a chart. Passing the cursor over a bar changes the color and displays a dollar amount.
- **Purchasing Projections** features a table that summarizes planned purchases. To add a new entry, click **Add Row**.



Planned Purchase Review

The Planned Purchase Review page contains two sections:

- **Vendor Analysis** is report of vendors.
- **Total Planned Purchases** is a data grid that displays a chart. Passing the cursor over a bar changes the color and displays a dollar amount.

Sample Websheet Application - AnyCo IT Department

[Language](#)
[Help](#)
[admin](#)
[Sign out](#)

View ▾
Create ▾
Edit ▾
Data Grid ▾
Administration ▾

Home >
Systems >
Planned Purchases Review >

✉
🖨
▶

▼ Vendor Analysis ✎ Edit

The following vendors have been shortlisted.

| Vendor | Name |
|--------------------------|--|
| iQuest PC | Laptops |
| Vireon Computers | Servers |
| UVA Network Solutions | Printers |
| Infian Soft | OS |
| Tetradigi Software | Project Management and Accounting Software |
| VirtualByte Technologies | InitCloud |

Control Panel ▾

- ➕ New Section
- ✎ Edit Sections
- ➕ New Page
- 📄 New Page as a Copy
- ✎ Edit Page
- 📁 Page Directory
- 📊 New Data Grid

Files + >

- 📄 vendors.txt ✎

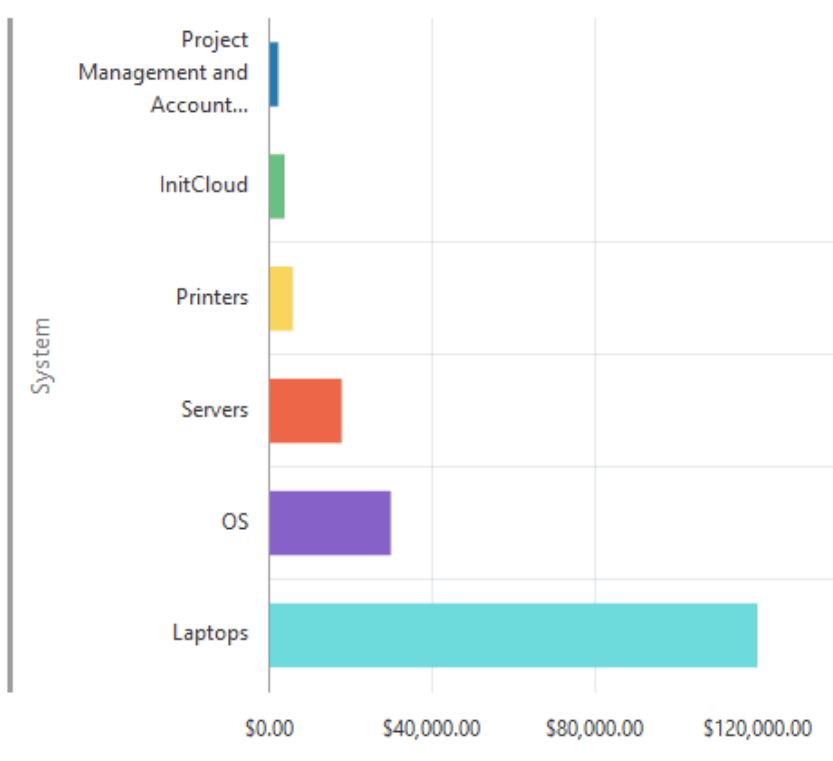
Tags + >

- 🏷 Discounts ✎
- 🏷 Pricing ✎
- 🏷 Purchases ✎

Notes + >

- 📝 The Total Planned ... ✎

▼ Total Planned Purchases ✎ Edit



| System | Total Planned Purchases (\$) |
|-----------------------------------|------------------------------|
| Laptops | ~115,000 |
| OS | ~35,000 |
| Servers | ~25,000 |
| Printers | ~10,000 |
| InitCloud | ~5,000 |
| Project Management and Account... | ~2,000 |

Systems Budget Review

The Systems Budget Review page contains two sections:

- **Annual Budget Allocation** is a report budget categories. To update report item, click the **Edit Row** icon.
- **Remaining Budget for the Year** is a data grid that displays a pie chart. Passing the cursor over a pie section displays the dollar amount.

Sample Websheet Application - AnyCo IT Department
Language Help admin Sign out

View Create Edit Data Grid Administration
Q Search Websheet

Home > Systems > Planned Purchases Review > Systems Budget Review
✉ 🖨️ ▶️

Annual Budget Allocation ✎ Edit ⌵

The following is the item-wise annual budget allocation:

| | System | Name | Annual Budget |
|---|----------|--|---------------|
| ✎ | Cloud | InitCloud | \$10,000.00 |
| ✎ | Hardware | Servers | \$85,000.00 |
| ✎ | Hardware | Laptops | \$172,000.00 |
| ✎ | Hardware | Printers | \$28,000.00 |
| ✎ | Software | Project Management and Accounting Software | \$15,000.00 |
| ✎ | Software | OS | \$62,000.00 |

For information on item-wise total costs incurred, see Systems Summary .

Control Panel ⌵

- 📄 New Section
- ✎ Edit Sections
- 📄 New Page
- 📄 New Page as a Copy
- ✎ Edit Page
- 📁 Page Directory
- 📄 New Data Grid

Files + >

No Files

Tags + >

- 📄 Annual Budget ✎
- 📄 System requirem... ✎

Notes + >

- 📄 If the Annual Bu... ✎
- 📄 The chart shows s... ✎

Remaining Budget for the Year ✎ Edit ⌵



| |
|-------------------------|
| Cloud - \$6,000.00 |
| Software - \$44,600.00 |
| Hardware - \$141,000.00 |

Modifying Websheet Content

Learn how to build a Websheet applications by studying a sample Websheet such as *AnyCo IT Department*.

- [Why Modify Sample Websheet Application?](#)
- [Updating a Websheet Section](#)

Why Modify Sample Websheet Application?

Analyzing the deconstruction of an application is an efficient way to speed up the learning process. If you break something, you can quickly delete the demonstration application and install it again.

Tip:

The App Gallery includes both *sample* and *productivity* applications. By default, sample applications are fully editable. In contrast, you must unlock a productivity application before you can edit it. Unlocking a productivity application makes it ineligible for future upgrades or support by Oracle Support.

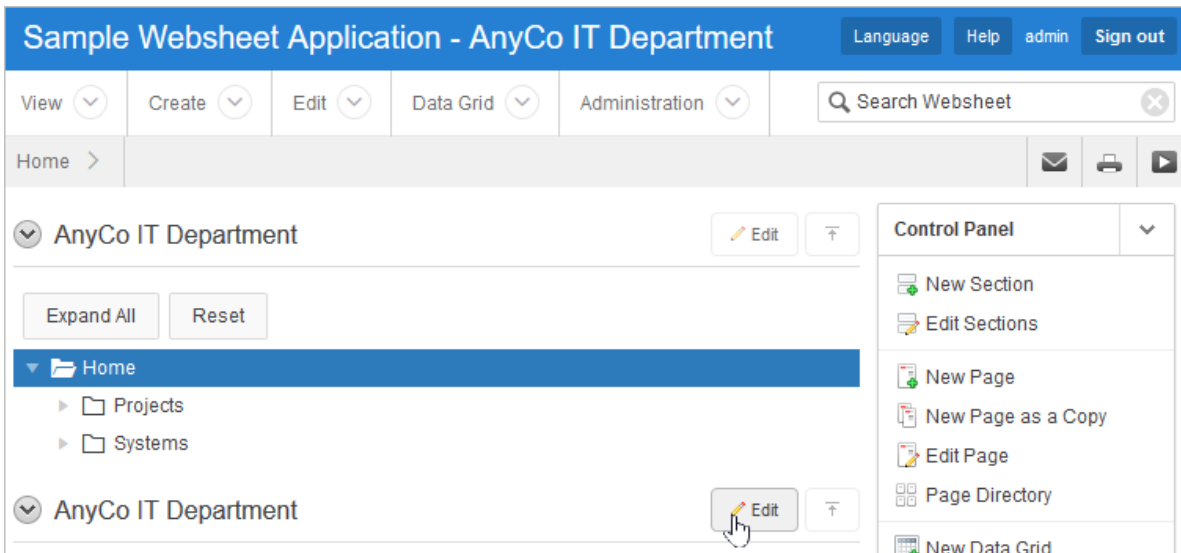
See Also:

["Creating Websheet Applications"](#)

Updating a Websheet Section

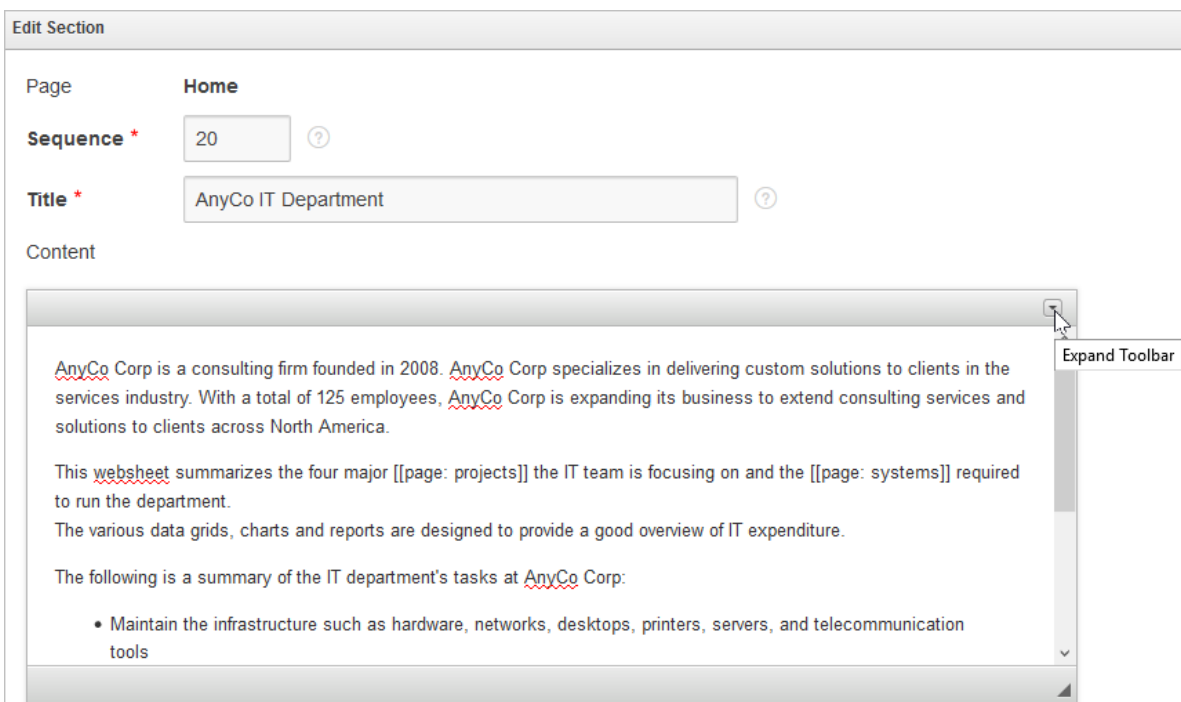
To update a Websheet section:

1. Run the Websheet.
2. Click the Edit icon that displays in the upper right of the section.



The Edit Section page appears. How you edit a specific section depends upon the section type.

3. Edit the fields provided.
4. To change the formatting, click the **Expand Toolbar** icon in the upper right corner.



The Toolbar appears. When you pass your cursor over an icon, a descriptive tooltip displays.

5. To hide the toolbar, click the **Collapse Toolbar** icon.

Edit Section

Page **Home**

Sequence * ?

Title * ?

Content

AnyCo Corp is a consulting firm founded in 2008. AnyCo Corp specializes in delivering custom solutions to clients in the services industry. With a total of 125 employees, AnyCo Corp is expanding its business to extend consulting services and solutions to clients across North America.

This websheet summarizes the four major [\[\[page: projects\]\]](#) the IT team is focusing on and the [\[\[page: systems\]\]](#) required to run the department.

The various data grids, charts and reports are designed to provide a good overview of IT expenditure.

The following is a summary of the IT department's tasks at AnyCo Corp:

- Maintain the infrastructure such as hardware, networks, desktops, printers, servers, and telecommunication tools

Collapse Toolbar

6. To save your changes, click **Apply Changes**.

Note:

"About App Gallery"

Utilizing App Administration

Administrators use App Administration to manage and monitor productivity and sample apps in their workspace.

- [Making a Service Request](#)
Create a service request for additional storage or terminate a workspace.
- [Setting Workspace Preferences](#)
Configure account login controls such as account expiration and locking, maximum login failures, and end user account lifetime (in days).
- [Managing Users](#)
Administrators in a Productivity and Sample Apps Only workspace can create new user accounts, manage existing user accounts, and change user passwords.
- [Managing Saved Interactive Reports](#)
View and delete saved interactive reports created by users of productivity and sample apps.

- [Managing Interactive Report Subscriptions](#)
View and manage interactive report subscriptions created by users of apps.
- [Viewing Activity Reports](#)
View a monthly calendar of app activity that details users and total page views by day.
- [Viewing Page Views](#)
View a report of app page views detailing the app, page, user, and elapsed time.
- [Viewing Top Users](#)
View a report of app page views aggregated by user.



See Also:

"Workspace and Application Administration" in *Oracle Application Express Administration Guide*

Making a Service Request

Create a service request for additional storage or terminate a workspace.



Tip:

Only users with administrator rights can access this functionality.

To make a service request for more storage or to terminate service:

1. Sign in to Oracle Application Express.
2. Click **App Gallery**.
3. Click **Administration**.
4. Click **Make a Service Request**.
5. To identify the type of service you would like to request, select one of the following:
 - Request Schema
 - Request Storage
 - Request Termination



Tip:

To see the amount of free space available in a workspace, expand **Tablespace Utilization** and select **Detailed Tablespace Utilization Report (may take several seconds)**.

6. Follow the on-screen instructions.

Setting Workspace Preferences

Configure account login controls such as account expiration and locking, maximum login failures, and end user account lifetime (in days).

 **Tip:**

Only users with administrator rights can access this functionality.

To set workspace preferences:

1. Sign in to Oracle Application Express.
2. Click **App Gallery**.
3. Click **Administration**.
4. Click **Set Workspace Preferences**.
5. Under Account Login Control:

 **Tip:**

To learn more about any attribute, see field-level Help.

- a. **Account Expiration and Locking** - Make a selection to determine whether Application Express end user accounts can be expired or locked. This feature applies only to end-user accounts created using the Application Express user management interface.
 - b. **Maximum Login Failures Allowed** - Enter a positive integer for the maximum number of consecutive unsuccessful authentication attempts allowed before an end-user account is locked. If you do not specify a value in this field, the instance-level setting for Maximum Login Failures Allowed is used.
 - c. **End User Account Lifetime (days)** - Enter a positive integer for the maximum number of days an end-user account password may be used before the account expires. If you do not specify a value in this field, the instance-level setting for Account Password Lifetime is used.
6. Click **Apply Changes**.

Managing Users

Administrators in a Productivity and Sample Apps Only workspace can create new user accounts, manage existing user accounts, and change user passwords.

A **Productivity and Sample Apps Only workspace** enables developers to share applications with other developers and administrators. It does not include any access to development tools such as App Builder or SQL Workshop. A Productivity and Sample Apps Only workspace includes the Apps page and the App Administration page.

 **Tip:**

Only users with administrator rights can access this functionality. This option is only available in a Productivity and Sample Apps Only workspace, but not in full workspace.

To manage users in a Productivity and Sample Apps Only workspace:

1. Sign in to the Productivity and Sample Apps Only workspace.
2. Click **Administration**.
3. Click **Manage Users**.

Managing users works the same in a Productivity and Sample Apps Only workspace as it does at the workspace-level.

 **See Also:**

"Managing Users in a Workspace" in *Oracle Application Express Administration Guide* and "[Using a Productivity and Sample Apps Only Workspace](#)"

Managing Saved Interactive Reports

View and delete saved interactive reports created by users of productivity and sample apps.

To manage saved interactive reports:

1. Sign in to Oracle Application Express.
2. Click **App Gallery**.
3. Click **Administration**.
4. Click **Saved Reports**.

Managing saved interactive reports works the same in a Productivity and Sample Apps Only workspace as it does at the workspace-level.

 **See Also:**

"Viewing and Deleting Saved Interactive Reports" in *Oracle Application Express Administration Guide*

Managing Interactive Report Subscriptions

View and manage interactive report subscriptions created by users of apps.

To manage interactive report subscriptions:

1. Sign in to Oracle Application Express.
2. Click **App Gallery**.
3. Click **Administration**.
4. Click **Subscriptions**.

Managing interactive report subscriptions works the same in a Productivity and Sample Apps Only workspace as it does at the workspace-level.

See Also:

"Managing Interactive Report Subscriptions" in *Oracle Application Express Administration Guide*

Viewing Activity Reports

View a monthly calendar of app activity that details users and total page views by day.

To view App Activity report:

1. Sign in to Oracle Application Express.
2. Click **App Gallery**.
3. Click **Administration**.
4. Click **App Activity**.

A monthly calendar of application activity appears.

Viewing Page Views

View a report of app page views detailing the app, page, user, and elapsed time.

To view the Page View report:

1. Sign in to Oracle Application Express.
2. Click **App Gallery**.
3. Click **Administration**.
4. Click **Page Views**.

Viewing Top Users

View a report of app page views aggregated by user.

To view the Top Users report:

1. Sign in to Oracle Application Express.
2. Click **App Gallery**.
3. Click **Administration**.
4. Click **Top Users**.

Using a Productivity and Sample Apps Only Workspace

A Productivity and Sample Apps Only workspace enables developers to share sample and productivity applications with other developers and administrators.



Note:

This information does not apply to Oracle Database Cloud Service (Database Schema).

- [What Is a Productivity and Sample Apps Only Workspace?](#)
- [Requesting a Productivity and Sample Apps Only Workspace](#)
- [Signing In To a Productivity and Sample Apps Only Workspace](#)
- [Using the App Page](#)

What Is a Productivity and Sample Apps Only Workspace?

A **Productivity and Sample Apps Only** workspace includes the App Gallery page and the App Gallery Administration page. It does not include any access to development tools such as App Builder or SQL Workshop. You can create a Productivity and Sample Apps Only workspace so that developers and administrators can share applications.



Tip:

End users cannot access a Productivity and Sample Apps Only workspace. End users can only run existing database or Websheet applications.

Requesting a Productivity and Sample Apps Only Workspace

To request a Productivity and Sample Apps Only workspace:

1. In a web browser, navigate to the Oracle Application Express Sign In page.
The Sign In page appears.
2. Under Workspace, click **Request a Workspace**.
The Request Service Wizard appears.
3. Select **Productivity and Sample Apps Only** and click **Next**.
4. Follow the on-screen instructions.

To learn more, see field-level Help.

 **Tip:**

You can manually control whether a workspace includes development tools, such as App Builder, by disabling access to development components. See "Configuring Workspace Preferences" and "Managing Component Availability for an Instance" in *Oracle Application Express Administration Guide*.

Signing In To a Productivity and Sample Apps Only Workspace

 **Tip:**

Sample applications are not available in a Productivity and Sample Apps Only workspace. Otherwise, the App Gallery page in a Productivity and Sample Apps Only workspace works the same as in a full development environment. See "[Managing Productivity and Sample Apps in a Full Development Environment](#)."

To sign in to a Productivity and Sample Apps Only workspace:

1. Sign in to the Productivity and Sample Apps Only workspace.
2. Click the **App Gallery** icon.
The App Gallery page appears.
3. Click **App Administration** to access the Application Administration page.

 **See Also:**

["Utilizing App Administration"](#)

Using the App Page

 **Tip:**

Sample applications are not available in a Productivity and Sample Apps Only workspace. Otherwise, the App Gallery page in a Productivity and Sample Apps Only workspace works the same as in a full development environment. See "[Managing Productivity and Sample Apps in a Full Development Environment](#)."

To access the App page:

1. Sign in to the Productivity and Sample Apps Only workspace.
2. Click the **App Gallery** icon.
The App Gallery page appears.
3. To search for an application, enter search terms in Search field.
4. Select an application to view.

The App Details page appears. Use this page to view information about an application, install an application, remove an application, or change the authentication of a previously installed application.

4

Managing the Application Life Cycle with Team Development

Team Development enables you to manage the development process by tracking new features, non-feature related tasks (or To Dos), bugs, and milestones. Users can also provide real-time feedback which then can be categorized into To Dos, bugs, or features.

- [Accessing Team Development](#)
Access Team Development by clicking the Team Development icon on the Workspace home page.
- [Team Development Home Page](#)
The Team Development home page is the starting point for tracking new features, non-feature related tasks (also known as To Dos), bugs, and milestones.
- [Tracking Features](#)
Track features from initial concept through implementation. You can organize features by release, assignee, tags, or associated milestones.
- [Tracking Milestones](#)
Track and manage important milestones associated with features, bugs, and To Dos.
- [Tracking To Dos](#)
Track To Dos (or assignments) associated with features and milestones.
- [Managing Bugs](#)
Track bugs (or software defects) associated with your application. You can assign bugs to developers, associate bugs with milestones, or track bugs by due date, status, and other attributes.
- [Managing Feedback](#)
Feedback provides a mechanism for end users to post general comments for application administrators and developers.
- [Managing Team Development Utilities](#)
Use Team Development Utilities to configure Team Development settings, view release summary information, enable support for file upload, access feature utilities, manage focus areas, update assignees, view uploaded files, purge data, manage news, and update links.

See Also:

"Managing Team Development" in *Oracle Application Express Administration Guide*

Accessing Team Development

Access Team Development by clicking the Team Development icon on the Workspace home page.

To access Team Development:

1. Sign in to Oracle Application Express.
The Workspace home page appears.
2. Click the **Team Development** icon.
The Team Development home page appears.



See Also:

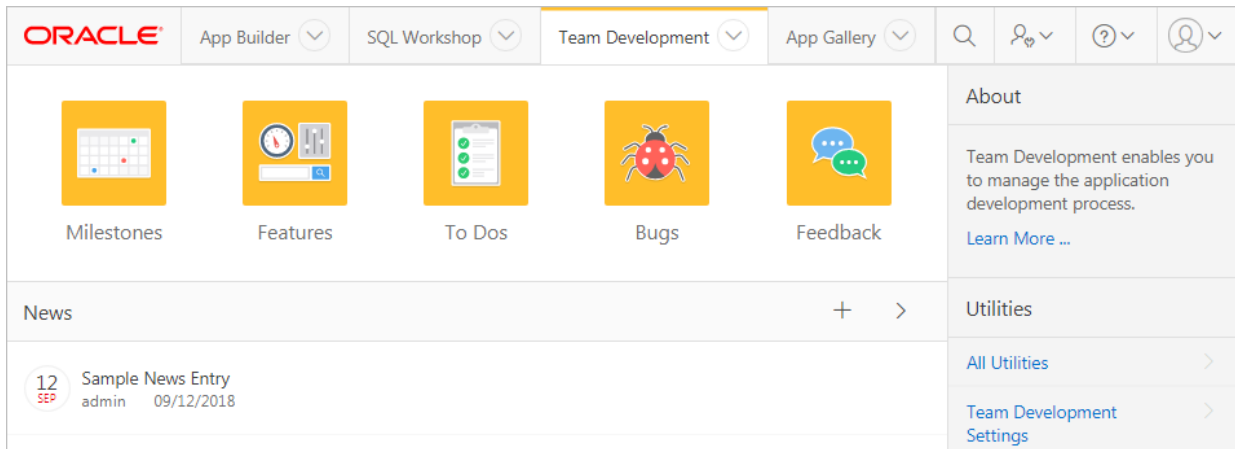
["Team Development Home Page"](#)

Team Development Home Page

The Team Development home page is the starting point for tracking new features, non-feature related tasks (also known as To Dos), bugs, and milestones.

The following large icons display on the Team Development home page:

- **Milestones.** Track events associated with the development process and associate milestones with features, bugs, and to dos.
- **Features.** Track features from initial concept through implementation. You can organize features by release, assignee, tags, or associated milestones.
- **To Dos.** Manage 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.
- **Bugs.** Track software defects or bugs. Bugs can be assigned, associated with milestones, and tracked by due date, status, and other attributes.
- **Feedback.** Gathering real-time comments, enhancement requests, and bugs from your application users.



News Region

Use the News region to communicate with other developers. You can add new or view news entries posted by other workspace users. News displays on the Application Express home page, the Team Development home page, and the News page.

Utilities Region

The Utilities region displays on the right side of the page and offers quick access to Team Development Utilities page. To access the Utilities page, click **All Utilities** or click a specific link.

Summary Regions

The bottom of the Team Development home page features the following summary regions:

- **Milestones.** Provides a summary of the number of days to the final release milestone.
- **Bugs.** Provides a summary of closed bugs.
- **Features.** Lists a percentage of features that are functionally complete.
- **To Dos.** Lists a percentage of completed To Dos.
- **Feedback.** Displays recent feedback entries.
- **Team Development Summary.** Lists the total number of features, to dos, milestones, bugs, and feedback entries.
- **Tags.** Displays a weighted list of all the tags associated with all Team Development components. Click the hyperlinks to link to a search page displaying all components with the selected tag. You can use the Tags summary to better manage Team Development components. To populate this summary, edit the Team Development component (that is, feature, To Do task, bug, and feedback entry and update the Tags field.

 **Tip:**

Most summary regions display a plus (+) sign and greater than sign (>) in the upper right corner. Click the plus (+) sign to add a component and click the greater than sign (>) to link to the component dashboard page.

 **See Also:**

["Managing News Entries"](#)

Tracking Features

Track features from initial concept through implementation. You can organize features by release, assignee, tags, or associated milestones.

- [Viewing the Features Report](#)
- [Creating a Feature](#)
- [About Feature Email Notifications](#)
- [Updating or Deleting a Feature](#)
- [About Creating Parent Features](#)
- [How to Attach Files to a Feature](#)
- [Viewing Other Feature Reports](#)
- [Assigning or Updating Milestones Associated with Features](#)

Viewing the Features Report

Use the Features report to view and edit existing features, create new features, or delete features.

To view features:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.

The Features Dashboard displays an overview of features defined within the workspace.

3. You can customize the display by making selections from the Release, Approval Status, Assignee, or Application lists and clicking **Set**.
4. Click the **Report** tab.

The Features Report page appears.

5. Customize the Features Report as you would any other interactive grid.
6. To view a feature, click the feature name.

**See Also:**

["Viewing Other Feature Reports"](#)

Creating a Feature

To create a feature:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.

The Features Dashboard displays an overview of features defined within the workspace.

3. Click **Create Feature**.

The Features dialog appears. Edit the appropriate fields.

4. Under Features:

Mandatory fields are marked with a red asterisk (*).

- **Feature** - Descriptive name of this feature.
 - **Tags** - Tags are keywords that further describe this feature. Separate multiple tags with commas. Tags display in a separate column in Report view.
 - **Owner** - Person responsible for this feature. To add values to this list, enter a name in the New Owner field.
 - **Contributor** - Another person who contributes to this feature. To add values to this list, enter a name in the New Contributor field.
 - **Focus Area** - Associates the feature to a specific focus area (or category). To add values to this list, enter a name in the New Focus Area field.
 - **Committed Feature** - Indicates whether the feature is a committed feature for the current release
 - **Approval Status** - Approval status of the feature. Indicates if the feature is to be implemented and the current progress. Approval Status options are preset and cannot be configured.
Approval Status only displays if enabled in Team Development Settings.
 - **Estimated Effort (in hours)** - Estimated time need to complete this feature.
 - **Progress Status** - Status of the feature. Indicates if the feature is to be implemented and the current progress. Status codes are preset and not configurable.
 - **Desirability** - The importance of the feature.
 - **Priority** - Priority or importance of the feature. Priorities are preset and not configurable.
 - **Parent** - Parent of this feature. Only features from the currently selected release will be display.
5. Under Dates:

- **Milestone** - Important events during this development cycle.
 - **Start Date** - Date feature implementation begins.
 - **Due Date / Date Completed** - Date feature implementation ends. If not provided, this date default to the Milestone date.
6. Under Summary:
 - **Publish this Feature** - Select **Yes** to make this feature viewable to a broader audience.
 - **Summary (displayed if Published)** - High level feature description designed to be published to a broad audience.
 7. Click **Create Feature**.



See Also:

["Renaming Focus Areas"](#) and ["Configuring Team Development Settings"](#)

About Feature Email Notifications

When you create a new features or update an existing feature, an email notification is sent to the feature owner and contributor and any assignees listed under User Interface, Testing, and Documentation.

About Subscribing and Unsubscribing to Features

Users who are not associated with a feature, can also receive email notifications by subscribing to the feature.

To subscribe or unsubscribe to a feature, go to the Features report, select the feature name, and click **Subscribe** or **Unsubscribe** as appropriate..



See Also:

["Viewing the Features Tree"](#)

Updating or Deleting a Feature

When you update an existing feature, an email notification is sent to the feature owner and contributor and any assignees listed under User Interface, Testing, and Documentation. Users who are not associated with a feature, can also receive email notifications by subscribing to the feature.

To update or delete a feature:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.

The Features Dashboard appears.

3. Click the **Report** tab.
The Report page appears.
4. To update a feature:
 - a. Click the feature name.
 - b. Edit the appropriate fields.
To learn more about an attribute, see field-level Help.
 - c. Some select lists are editable. To add new values to editable select lists, enter a value in the field with the title *New*.
 - d. Click **Apply Changes**.
5. To delete a feature:
 - a. Click the feature name.
 - b. Click **Delete**.

**See Also:**

["Renaming Focus Areas"](#)

About Creating Parent Features

You can group features hierarchically using the Parent Feature attribute. Selecting a Parent Feature creates a parent/child relationship. When viewing features in Tree view, features are grouped first by release and then by parent.

**See Also:**

["Viewing the Features Tree"](#)

How to Attach Files to a Feature

If enabled at the workspace-level, you can attach files to features.

- [Attaching a File to a Feature](#)
- [Viewing, Editing, and Removing Attached Files](#)

**See Also:**

"[Downloading File Attachments](#)" and "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*.

Attaching a File to a Feature

To use this feature, you must set the Enable File Repository attribute to **Yes** in Administration, Manage Service, Set Workspace Preferences.

To attach a file to a feature:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.
The Features Dashboard appears.
3. Click the **Report** tab.
The Report page appears.
4. Click the feature name.
5. Click the **Files** tab.
6. Click **Add File**.
The Add File dialog appears.
7. Locate the file to be attached and click **Add File**.

Viewing, Editing, and Removing Attached Files

To view, edit, or remove attached files:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.
The Features Dashboard appears.
3. Click the **Report** tab.
The Report page appears.
4. Click the feature name.
5. Click the **Files** tab.
6. Click **View and Manage Files**.
The Files page appears.
7. To download the file, click the name.
8. To edit the file description:
 - a. Click the **Edit** icon.
 - b. Update the description and click **Apply Changes**.
9. To delete an attached file:
 - a. Click the **Edit** icon.
 - b. Click **Delete**. When prompted, confirm your selection.

Viewing Other Feature Reports

Click the tabs at the top of the Features page to view other reports including Tree, Calendar, History Progress Log, Focus Areas, and Owners.

- [Viewing the Features Tree](#)
- [Viewing the Features Calendar](#)
- [Viewing The Feature History Report](#)
- [Viewing the Feature Progress Log](#)
- [Viewing the Feature Focus Areas Chart](#)
- [Viewing the Approval Statuses](#)
- [Viewing the Feature Owners Chart](#)

 **See Also:**

- ["Viewing the Features Report"](#)
- ["Assigning or Updating Milestones Associated with Features"](#)
- ["Accessing Team Development Utilities"](#)

Viewing the Features Tree

To view the Features tree:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.

The Features Dashboard appears.

3. Click the **Tree** tab.

The Tree page displays features in a hierarchical tree by release.

Features are grouped first by release and then by parent. To expand a grouping, click the small triangle to the left of the feature name.

 **Tip:**

When displaying features by a given release, features with parent or children features in different releases do not display.

4. Controls at the top of the page include:
 - **Release** - To select a new release, make a selection from the Release list and click **Set**.
 - **Include Percent Complete?** - The feature number and percent complete display to the right of the feature name. To hide or display the percent complete, select **Yes** or **No** from Include Percent Complete and click **Set**.
 - **Reset** - Resets the report to the default view.
 - **Collapse All** - Collapses the tree.
 - **Expand All** - Expands the tree.

- **Create Feature** - Displays the Feature dialog enabling you to create a new feature.
5. To edit a feature, select it. When the Feature dialog appears, edit the appropriate fields and click **Apply Changes**.
 6. To delete a feature, select it. When the Feature dialog appears, click **Delete**.

Viewing the Features Calendar

To view the Features Calendar:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.

The Features Dashboard appears.

3. Click the **Calendar** tab.

The Calendar page displays features by due date or date completed in a calendar format.

If a parent features have been defined, the due date or date completed displays in a calendar format.

4. Controls at the top of the page include:
 - **Release** - To select a new release, make a selection from the Release list and click **Set**.
 - **Assignee** - Select an assignee and click **Set**.
 - **Reset** - Resets the Calendar to the default view.
 - **Create Feature** - Displays the Feature dialog enabling you to create a new feature.
 - **Previous/Next** - View to the previous month or the next month.
 - **Month** - View the calendar by month.
 - **List** - View the calendar entries in a list format by day.
5. To edit a feature, select it. When the Feature dialog appears, edit the appropriate fields and click **Apply Changes**.
6. To delete a feature, select it. When the Feature dialog appears, click **Delete**.

Viewing The Feature History Report

To view the Feature History report:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.

The Features Dashboard appears.

3. Click the **History** tab.

The Feature History report displays an interactive report of recently changed features, including the old value and new value and who made the update.

4. To customize the report, use the Actions menu in Search bar at the top of the page. Click **Reset** return the report to the default view.

5. To view a specific feature, click the feature name.

**See Also:**

["Customizing Interactive Reports in a Running Application"](#) and ["About the Actions Menu"](#)

Viewing the Feature Progress Log

Developers can track completed tasks associated with a feature under Progress.

To view the Feature Progress Log:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.
The Features Dashboard appears.
3. Click the **Progress Log** tab.
The Progress Log page displays as an interactive report of progress log entries.
4. To customize the report, use the Actions menu in Search bar at the top of the page. Click **Reset** return the report to the default view.
5. To edit a log entry, click the **Edit** icon to the left of the Progress Note.

**See Also:**

["Customizing Interactive Reports in a Running Application"](#)

Viewing the Feature Focus Areas Chart

The Focus Areas page displays a chart of the number of features per defined focus area by release.

To view the Feature Focus Areas chart:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.
The Features Dashboard appears.
3. Click the **Focus Areas** tab.
The Focus Areas page displays a chart of the number of features per defined focus area by release.
4. To customize the report, use the Actions menu in Search bar at the top of the page. Click **Reset** return the report to the default view.
5. To rename a Focus Area:
 - a. Click **Manage Focus Areas**.

- b. Locate the Focus Area and click the **Edit** icon.
- c. In Rename To, enter your edits and click **Apply Changes**.

**See Also:**

["Renaming Focus Areas"](#)

Viewing the Approval Statuses

To view Features Approval Statuses:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.
The Features Dashboard appears.
3. Click the **Approval Statuses** tab.

The Approval Statuses page displays the status and description . To create a new status, click **Create**.

Viewing the Feature Owners Chart

To view the Features Owner chart:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.
The Features Dashboard appears.
3. Click the **Feature Owners** tab.
The Feature Owners page displays a chart of the number of features by owner.
4. To alter the chart, select another Release, Minimum Status, or Maximum Status, and click **Set**. To return to the default chart, click **Reset**.

Assigning or Updating Milestones Associated with Features

Use Select a Utility to assign milestones to features, set feature due dates to milestone dates, change milestones for overdue features, or push due dates of open features.

To access Select a Utility:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.
The Features Dashboard appears.
3. Click the **Utilities** tab.
4. Select one of the following and click **Next**.
 - **Assign milestones to features (only those with no milestone currently assigned).**

- **Set feature due date to milestone date (only affects features).**
 - **Push due date of open features.**
5. Follow the on-screen instructions.

Tracking Milestones

Track and manage important milestones associated with features, bugs, and To Dos.

- [Viewing the Milestones Report](#)
- [Creating a Milestone](#)
- [Updating or Deleting a Milestone](#)
- [Viewing Other Milestone Reports](#)

Viewing the Milestones Report

To view milestones:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Milestones** icon.

The Milestones Dashboard appears.

The Milestones Dashboard displays an overview of milestones within the workspace. You can filter by release or link to a specific milestone or milestones by owner.

3. Click the **Report** tab.

The Milestones Report page displays as an interactive report. You can customize the appearance of page using the Search bar at the top of the page.



See Also:

["Customizing Interactive Reports in a Running Application"](#)

Creating a Milestone

To create a milestone:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Milestones** icon.
The Milestones Dashboard page appears.
3. Click **Create Milestone**.
4. Fill in the appropriate fields. Mandatory fields are marked with a red asterisk (*).
To learn more about an attribute, see field-level Help.
5. To add new values to editable select lists, enter a value in the field with the title *New*. For example, to add name to the Owner list, enter a value in New Owner.

The value you enter is added.

6. Click **Create Milestone**.

Updating or Deleting a Milestone

To update a milestone:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Milestones** icon.

The Milestones Dashboard page appears and displays the following regions:

3. Click the **Report** tab.

The Milestones Report page appears.

4. Select the milestone.
5. Click **Edit Milestone**.
6. Edit the appropriate fields.

To learn more about an attribute, see field-level Help.

7. Click **Apply Changes**.

Viewing Other Milestone Reports

Click the tabs at the top of the Milestones page to view other reports including Calendar, By Owner, or Features by Milestone.

- [Viewing the Milestone Calendar](#)
- [Viewing the Milestones By Owner Report](#)
- [Viewing the Features by Milestone Report](#)



See Also:

["Accessing Team Development Utilities"](#)

Viewing the Milestone Calendar

To view the Milestone Calendar:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Milestones** icon.

The Milestones Dashboard appears.

3. Click the **Calendar** tab.

The Calendar page displays milestones by date in a calendar format.

4. Controls at the top of the page include:
 - **Release** - Restrict report to a given release, or display all releases and click **Set**.

- **Create Milestone** - Displays the Milestone dialog enabling you to create a new milestone.
- **Previous/Next** - View to the previous month or the next month.
- **Month** - View the calendar by month.
- **List** - View the calendar entries in a list format by day.

Viewing the Milestones By Owner Report

To view the Milestones By Owner report:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Milestones** icon.
The Milestones Dashboard appears.
3. Click the **By Owner** tab.
To alter the report, select another release or milestone and click **Set**. To return to the default chart, click **Reset**.
4. To alter the report, select another release or milestone and click **Set**.
5. To return to the default chart, click **Reset**.
6. To edit a milestone, click **Edit Milestone**.

Viewing the Features by Milestone Report

To view the Features by Milestone report:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Milestones** icon.
The Milestones Dashboard appears.
3. Click the **Features by Milestone** tab.
The Feature by Milestone report displays as an interactive report and displays features associated with each milestone.
4. To customize the report, use the Actions menu in Search bar at the top of the page. Click **Reset** return the report to the default view.
5. To view a specific feature, click the feature name.



See Also:

["Customizing Interactive Reports in a Running Application"](#) and ["About the Actions Menu"](#)

Tracking To Dos

Track To Dos (or assignments) associated with features and milestones.

To Dos are work assignments associated with features but not directly related to software functionality. You can assign ToDos to developers and organize them by status, release, assignee, or application.

- [Viewing To Dos](#)
- [Creating a To Do](#)
- [Updating a To Do](#)
- [Managing Files Attached to a To Do](#)
- [Viewing Other To Do Reports](#)

Viewing To Dos

To view To Dos:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **To Dos** icon.
The To Dos Dashboard appears. The To Dos Dashboard displays an overview of To Dos within the workspace. You can filter by release, assignee, or application.
3. Click the **Report** tab.
The To Dos Report page appears.
4. Customize the report as you would any other interactive grid.
5. To create a new To Do, click **Create To Do** in the upper right corner.



See Also:

"Using Interactive Grids" in *Oracle Application Express End User's Guide* and "[Updating a To Do](#)"

Creating a To Do

To create a To Do:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **To Dos** icon.
The To Dos dashboard appears.
3. Click **Create To Do**.
4. Fill in the appropriate fields. Mandatory fields are marked with a red asterisk (*).
To learn more about an attribute, see field-level Help.
5. To add new values to editable select lists, enter a value in the field with the title *New*. For example, to add name to the Assigned To list, enter a value in New Assignee.
The value you enter is added.
6. Click **Create To Do**.

Updating a To Do

To update a To Do:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **To Dos** icon.
The To Dos Dashboard appears.
3. Click the **Report** tab.
The To Dos Report page appears.
4. Customize the report as you would any other interactive grid.
5. Select a To Do.
6. Edit the appropriate fields.
7. Click **Apply Changes**.



See Also:

"Using Interactive Grids" in *Oracle Application Express End User's Guide*

Managing Files Attached to a To Do

If enabled at the workspace-level, you can attach files to a To Do.

- [Attaching a File to a To Do](#)
- [Viewing, Editing, and Removing Attached Files](#)



See Also:

"[Downloading File Attachments](#)" and "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*.

Attaching a File to a To Do

To use this feature, you must set the Enable File Repository attribute to **Yes** in Administration, Manage Service, Set Workspace Preferences.

To attach a file to a To Do:

1. Navigate to the To Do:
 - a. On the Workspace home page, click the **Team Development** icon.
 - b. Click the **To Dos** icon.
The To Dos Dashboard appears.

- c. Click the **Report** tab.
 - d. Select a To Do.
 2. From Actions, click **Add File**.
 3. On Add File:
 - a. File - Locate the file to be uploaded.
 - b. Description - Enter a description of the file.
 - c. Click **Add File**.

A new region, called Files, displays at the bottom of the page.

Viewing, Editing, and Removing Attached Files

To manage attached files:

1. Navigate to the To Do:
 - a. On the Workspace home page, click the **Team Development** icon.
 - b. Click the **To Dos** icon.

The To Dos Dashboard appears.
 - c. Click the **Report** tab.
 - d. Select a To Do.
2. Under Files, click **View and Manage Files**.

The Files page appears.
3. To download the file, click the name.
4. To edit the file description:
 - a. Click the **Edit** icon.
 - b. Update the description and click **Apply Changes**.
5. To delete an attached file:
 - a. Click the **Edit** icon.
 - b. Click **Delete**. When prompted, confirm your selection.

Viewing Other To Do Reports

Click the tabs at the top of the To Dos page to view other reports including Calendar and Progress Log.

- [Viewing the To Dos Calendar](#)
- [Viewing the To Do Progress Log](#)

Viewing the To Dos Calendar

To view the To Dos Calendar:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **To Dos** icon.

The To Dos Dashboard appears.

3. Click the **Calendar** tab.

The To Dos page displays features by due date or date completed in a calendar format.

The To Dos Calendar appears.

4. Controls at the top of the page include:
 - **Release** - To select a new release, make a selection from the Release list and click **Set**.
 - **Reset** - Resets the Calendar to the default view.
 - **Create To Do** - Displays the To Do page enabling you to create a new feature.
 - **Previous/Next** - View to the previous month or the next month.
 - **Month** - View the calendar by month.
 - **List** - View the calendar entries in a list format by day.
5. To edit a To Do, select it. When the To Do appears, edit the appropriate fields and click **Apply Changes**.
6. To edit a To Do, select it. When the To Do appears, click **Delete**.

Viewing the To Do Progress Log

To view the To Do Calendar:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **To Dos** icon.

The To Dos Dashboard appears.
3. Click the **Progress Log** tab.

The Progress Log page displays an interactive report of progress log entries. To customize the report, use the Search bar at the top of the page.

4. To edit an entry, click the Progress Entry name.



See Also:

["Customizing Interactive Reports in a Running Application"](#)

Managing Bugs

Track bugs (or software defects) associated with your application. You can assign bugs to developers, associate bugs with milestones, or track bugs by due date, status, and other attributes.

- [Viewing Bugs](#)
- [Creating a Bug](#)
- [Editing a Bug](#)
- [Managing Files Attached to a Bug](#)

- [Viewing Other Bug Reports](#)

Viewing Bugs

To view bugs:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears and displays an overview of known software defects.
3. Click the **Report** tab.
The Bugs Report page appears. Customize the appearance of the page using the Search bar at the top of the page.
4. To edit a bug, click the bug title.



See Also:

["Customizing Interactive Reports in a Running Application"](#) and ["Editing a Bug"](#)

Creating a Bug

To create a bug:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears.
3. Click **Create Bug**.
4. Fill in the appropriate fields. Mandatory fields are marked with a red asterisk (*).
To learn more about an attribute, see field-level Help.
5. To add new values to editable select lists, enter a value in the field with the title *New*.
For example, to add a name to the Assigned To list, enter a value in **New Assignee**.
6. Click **Create Bug**.

Editing a Bug

To edit a bug:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears and displays an overview of known software defects.

3. Click the **Report** tab.
The Bugs Report page appears. Customize the appearance of the page using the Search bar at the top of the page.
4. Select a bug.
5. Edit the appropriate fields.
To learn more about an attribute, see field-level Help.
6. Click **Apply Changes**.

Managing Files Attached to a Bug

If enabled at the workspace-level, you can attach files to a bug.

- [Attaching a File to a Bug](#)
- [Viewing, Editing, and Removing Attached Files](#)

See Also:

"[Downloading File Attachments](#)" and "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*.

Attaching a File to a Bug

To use this feature, you must set the Enable File Repository attribute to **Yes** in Administration, Manage Service, Set Workspace Preferences.

To attach a file to a bug:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears and displays an overview of known software defects.
3. Click the **Report** tab.
The Bugs Report page appears. Customize the appearance of the page using the Search bar at the top of the page.
4. Click the bug title.
5. Under Actions, click **Add File**.
6. In Add File:
 - a. File - Locate the file to be uploaded.
 - b. Description - Enter a description of the file.
 - c. Click **Add File**.

A new region, called Files, displays at the bottom of the page.

Viewing, Editing, and Removing Attached Files

To manage attached files:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears and displays an overview of known software defects.
3. Click the **Report** tab.
The Bugs Report page appears. Customize the appearance of the page using the Search bar at the top of the page.
4. To edit a bug, click the bug title.
5. Scroll down to Files.
6. Click **View and Manage Files**.
The Files page appears.
7. To download the file, click the name.
8. To edit the file description:
 - a. Click the **Edit** icon.
 - b. Update the description and click **Apply Changes**.
9. To delete an attached file:
 - a. Click the **Edit** icon.
 - b. Click **Delete**. When prompted, confirm your selection.

Viewing Other Bug Reports

Click the tabs at the top of the Bugs page to view other reports including Calendar, By Developer, and By Day.

- [Viewing the Bug Calendar](#)
- [Viewing the Bugs By Developer](#)
- [Viewing the Bugs By Day](#)

Viewing the Bug Calendar

To view the Bug Calendar:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears.
3. Click the **Calendar** tab.
The Calendar appears.
4. Controls at the top of the page include:
 - **Show** - Restrict the report to include open or closed comments.

- **Release** - Selects a new release. Make a selection from the Release list and click **Set**.
- **Assigned To** - Constrains the report to display only bugs associated with the selected developer. Select a developer and click **Set**.
- **Reset** - Resets the Calendar to the default view.
- **Create Bug** - Creates a new bug.
- **Previous/Next** - View to the previous month or the next month.
- **Month** - View the calendar by month.
- **List** - View the calendar entries in a list format by day.

Viewing the Bugs By Developer

To view the Bugs By Developer:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears.
3. Click the **By Developer** tab.
The Bugs by Developer report appears.
4. Controls at the top of the page include:
 - **Show** - Restrict the report to include open or closed comments.
 - **Release** - Selects a new release. Make a selection from the Release list and click **Set**.
 - **Reset** - Resets the Calendar to the default view.
 - **Create Bug** - Creates a new bug.

Viewing the Bugs By Day

To view the Bugs By Day:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
The Bugs Dashboard appears.
3. Click the **By Day** tab.
The Bugs by Developer report appears.
4. Controls at the top of the page include:
 - **Release** - Selects a new release. Make a selection from the Release list and click **Set**.
 - **Reset** - Resets the Calendar to the default view.
 - **Create Bug** - Creates a new bug.

Managing Feedback

Feedback provides a mechanism for end users to post general comments for application administrators and developers.

- [About Feedback](#)
Adding Feedback enables application administrators and developers to gather real-time comments, enhancement requests, and bugs from application users. Feedback enhances communication between the user community and the application owners.
- [Adding Feedback Using the Create Application Wizard](#)
Add a Feedback page when running the Create Application Wizard.
- [Adding Feedback Using the Create Page Wizard](#)
Use the Create Page Wizard to add Feedback to an existing application.
- [Submitting Feedback Within a Running Application](#)
Users can submit feedback by clicking the Feedback navigation bar icon in the upper right of a running application.
- [Reviewing Feedback Within an Application](#)
View feedback within a running application by clicking User Feedback on the Administration page.
- [Configuring Feedback Within an Application](#)
Configure feedback within a running application by clicking Feedback Settings on the Administration page.
- [Tracking Feedback Team in Development](#)
Track all feedback within a workspace in Team Development.

About Feedback

Adding Feedback enables application administrators and developers to gather real-time comments, enhancement requests, and bugs from application users. Feedback enhances communication between the user community and the application owners.

Feedback provides a mechanism for end users to post general comments for application administrators and developers. The posts include useful session state information to help developers determine where the end user sent the feedback from. Providing such an easy feedback loop enhances communication between the user community and the application owners, which leads to greatly improved end user satisfaction.

Creating Feedback:

- Adds a Feedback navigation bar icon in the running application which users can click to leave feedback.
- Creates an report for viewing and updating feedback.
- Captures the application and page ID, feedback comments, date and time, and user information.

Choosing How to Add Feedback

You can create a Feedback page automatically using a wizard when you create a new application, or by adding a page to an existing application. In both the Create

Application Wizard and the Create Page Wizard, you simply select the Feature, **Feedback**.

How Customers Submit Feedback

Users can submit feedback by clicking the **Feedback** navigation bar icon in the upper right of a running application. In the Feedback window, they can rate their user experience by selecting an emoji, entering feedback comments, or uploading files.

Viewing Feedback within an Application

Administrators and developers can track and manage Feedback at the application-level, by running the application and accessing Feedback region on the Administration page. To control whether users can upload attachments, click **Feedback Settings**. To view feedback entries, select **User Feedback** to view the Manage Feedback report.

Tracking Feedback in Team Development

Administrators and developers can track all feedback within a workspace in Team Development, Feedback. Each feedback entry tracks the application and page ID and name, displays any feedback comments. Administrators and developers can assign a status and classify the information as a Bug, To Do, or Feature.

Adding Feedback Using the Create Application Wizard

Add a Feedback page when running the Create Application Wizard.

To create a database application which includes a Feedback page:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Click **Application**.
4. For Name, enter the name used to identify the application to developers.
5. For Appearance, accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
6. To add a new page (such as a blank page, calendar, chart, form, report, and so on) click **Add Page** and select the desired page type.
7. Under Features, select **Feedback**.
8. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
9. Click **Create Application**.

See Also::

["Understanding Page Types, Features, and Settings"](#)

Adding Feedback Using the Create Page Wizard

Use the Create Page Wizard to add Feedback to an existing application.

Tip:

This section describes how to create a feedback by running the Create Page Wizard from the Application home page. You can also run this wizard in Page Designer by clicking the **Create** menu and selecting **Page**.

To add a feedback page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
The Application home page appears.
3. Click the **Create Page** button.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page.
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - To create a report, select **Feature** and then **Feedback**.

Tip:

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

The Create Feedback Page appears.

5. On the Create Feedback Page:
 - a. Starting Page Number - Enter an unused starting page number. This feature will create multiple pages starting with the number specified.
 - b. Page Group - Identify the name of the page group to associate with this page. Page groups help developers manage the pages within an application. To use an existing page group, select the name from the list.
 - c. Build Option - The listed build option(s) to be created to support this feature.
 - d. Include in Navigation Bar - Select **Yes** to create a navigation bar entry within the current application to invoke your new feedback page.
 - e. Navigation Bar Label - Enter the name of the new navigation bar entry used to access the new feedback page.
 - f. Administration Page Preference - Specify the administration page to which a list will be added for this feature.

Options include:

- **Create a new page**
- **Identify an existing page**

Depending upon your selection, the UI changes.

- g. Follow the on-screen instructions.
6. Click **Create**.

Submitting Feedback Within a Running Application

Users can submit feedback by clicking the Feedback navigation bar icon in the upper right of a running application.

To submit feedback managed with the application:

1. Run the application for which you want to submit feedback.
2. Click the **Feedback** navigation bar icon in the upper right corner.
A Feedback window appears.
3. In Feedback:
 - a. Experience - Select an emoji that reflects your user experience (**Positive**, **Neutral**, or **Negative**).
 - b. Feedback - Enter a comment.
 - c. Attachment - Click **Choose File** to upload an attachment.

Tip:

Administrators can enable and disable attachments by selecting **Administration** and editing **Feedback Settings**.

4. Click **Submit Feedback**.

Reviewing Feedback Within an Application

View feedback within a running application by clicking User Feedback on the Administration page.

To view feedback within an application:

1. Run the application.
2. Click **Administration** in the left navigation bar.
3. Click **User Feedback**.
The Manage Feedback page displays an interactive report.
4. Click the Edit icon adjacent to the entry you wish to view.
5. On the Feedback page
 - a. Review the user feedback.
 - b. Response - Respond to the feedback.

- c. Status - Select one of the following:
 - **No Action**
 - **Acknowledged**
 - **Open**
 - **Closed**
6. To delete the feedback, click **Delete**.
7. To save your changes, click **Apply Changes**.

Configuring Feedback Within an Application

Configure feedback within a running application by clicking Feedback Settings on the Administration page.

Developer and administrators can also configure Feedback Settings that control support for attachments.

To configure Feedback Settings:

1. Run the application.
2. Click **Administration** in the left navigation bar.
3. Click **Feedback Settings** .
Feedback Settings appears.
4. For Enable Attachments, select **Yes** to enable users to upload attachments or **No** to disable this functionality.
5. To save your changes, click **Apply Changes**.

Tracking Feedback Team in Development

Track all feedback within a workspace in Team Development.

Administrators and developers can track all feedback within a workspace in Team Development. Each feedback entry tracks the application and page ID and name, displays any feedback comments. Administrators and developers can assign a status and classify the information as a Bug, To Do, or Feature.

- [Accessing the Feedback Report in Team Development](#)
- [Viewing Feedback by Filing User](#)
- [Viewing the Feedback Calendar](#)

Accessing the Feedback Report in Team Development

To access the feedback report in Team Development:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Feedback** icon.
The Feedback Dashboard appears.
3. Click the **Report** tab.

4. Select a feedback number.
The View/Edit Feedback dialog appears.
5. On View/Edit Feedback:
 - a. Review the user feedback.
 - b. Type - Classify the feedback. Select one of the following:
 - **General Comment**
 - **Enhancement Request**
 - **Bug**
 - c. Disposition - Determine how to log the feedback. Select one of the following:
 - **Log as Bug.**
Enter the appropriate information and click **Create Bug**.
 - **Log as To Do.**
Enter the appropriate information and click **Create To Do**.
 - **Log as Feature.**
Enter the appropriate information and click **Create Feature**.
6. For Status, select a status of the feedback entry:
 - **0. No Status**
 - **1. Acknowledged**
 - **2. Additional information requested**
 - **3. Open Processing Feedback**
 - **4. Closed**
7. To delete a feedback entry, click **Delete**.
8. To save your changes, click **Apply Changes**.

Viewing Feedback by Filing User

To view feedback by filing user:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Feedback** icon.
The Feedback Dashboard appears.
3. Click the **By Filing User** tab.
The By Filing User report appears.
4. To customize the report, use the Actions menu in Search bar at the top of the page. Click **Reset** return the report to the default view.
5. Click user under Filing User to view the associated feedback.

Viewing the Feedback Calendar

To view the Feedback Calendar:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Feedback** icon.
The Feedback Dashboard appears.
3. Click the **Calendar** tab.
The Calendar appears.

Managing Team Development Utilities

Use Team Development Utilities to configure Team Development settings, view release summary information, enable support for file upload, access feature utilities, manage focus areas, update assignees, view uploaded files, purge data, manage news, and update links.

The Utilities region displays on the right side of the Team Development home page.

- [Accessing Team Development Utilities](#)
Use Team Development Utilities to manage team development data. You can access Team Development Utilities from either the Utilities region on the Team Development home page or from Utilities page.
- [Configuring Team Development Settings](#)
Configure Team Development Settings to specify default behavior for features and To Dos. Choose default values for the release, owner, and priority, and determine what feature attributes are enabled for the current workspace.
- [Viewing a Release Summary](#)
View a summary of report by release or milestones on the Release Summary page.
- [Managing Feature Utilities](#)
Use the Feature Utilities page to assign milestones to features, set feature due dates to milestone dates, or push due dates of open features.
- [Renaming Focus Areas](#)
When you create or update a feature, you can assign it to a focus area. Use Manage Focus Areas, to rename a focus area.
- [Downloading File Attachments](#)
If enabled at the workspace-level, you can attach files to a feature, To Do, or bug. Use the Team Development Files page to download attached files.
- [Purging Data](#)
Remove data associated with a Team Development component type (that is, features, milestones, To Dos, bugs, or feedback).
- [Managing News Entries](#)
Use the News region to communicate with other users in the current workspace. News displays on the Application Express home page, the Team Development home page, and News page.
- [Managing Links with Workspace Members](#)

 **See Also:**

"Managing Team Development" in *Oracle Application Express Administration Guide*

Accessing Team Development Utilities

Use Team Development Utilities to manage team development data. You can access Team Development Utilities from either the Utilities region on the Team Development home page or from Utilities page.

To access Team Development Utilities:

1. On the Workspace home page, click the **Team Development** icon.
2. Note the Utilities region on the right side of the page.

 **Tip:**

You can access most utilities by either clicking links in the Utilities region or by accessing the Utilities page.

3. From the Utilities region on the right side of the page, click **All Utilities**.
4. The Utilities page features the following links:
 - **Team Development Settings** - Specify how To Dos and bugs display, configure default values for release, owner, and priority, and determine what tracking attributes are enabled for the current workspace.
 - **Release Summary** - Review workspace and developer statistics.
 - **Disable Files/Enable Files** - Links to an administration page that where you can enable or disable support for file upload. The text that displays changes based on the current configuration.
 - **Feature Utilities** - Manage milestones for multiple features at once. Use this page to assign milestones to features, set the feature due date to a milestone date, change milestone for overdue features, and push due dates for open features.
 - **Manage Focus Areas** - Manage focus areas that display on the Feature page.
 - **Update Assignees** - Enables you to reassign selected components. Only incomplete components can be reassigned (that is, those components having a status less than 100% or milestone date in the future).
 - **View Files** - Manage uploaded files attached to features, To Dos, and bugs.

This option toggles between **View Files** and **Enable Files** depending upon whether your administrator has enabled file upload capability. If **Enable Files** displays, then file upload capability has not yet been enabled by your administrator.

- **Purge Data** - Delete all entries for the selected components. This is useful when beginning a new development cycle or after using a workspace for testing.
- **Manage News** - Manage news entries that display on the Team Development home page and Workspace home page.
- **Manage Links** - Manage links to share with other workspace users.
- **Auto Create To Dos** - Automatically create a To Do for every page in an application.
- **Push Past Due Bugs** - Extend the estimated fix date of open bugs. You can filter and then select just the bugs you want to push and then extend for any number of days.

 **See Also:**

- ["Configuring Team Development Settings"](#)
- "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*

Configuring Team Development Settings

Configure Team Development Settings to specify default behavior for features and To Dos. Choose default values for the release, owner, and priority, and determine what feature attributes are enabled for the current workspace.

To edit Team Development settings:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Team Development Settings**.
The Settings page appears.
3. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help.
 - a. Release - Default release used when creating new tasks.
 - b. Owner - Default owner when creating features and To Do tasks.
 - c. Priority - Default priority when creating features.
 - d. Enable Approval Status - Enables or disables feature approval status.
4. Under, Track Attributes specify which attributes are available. For each group of attributes, select **Yes** to include the group or **No** to hide the group.
 - a. **User Interface**
 - b. **Testing**
 - c. **Documentation**
 - d. **Globalization**
 - e. **Security**

- f. **Accessibility**
5. Click **Apply Changes**.

 **Tip:**

Click **Reset to Default** to return the settings to the default values.

Viewing a Release Summary

View a summary of report by release or milestones on the Release Summary page.

To access the Release Summary report:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Release Summary**.
The Release Summary page appears.
3. Use the Search bar at the top of the page to filter the view.
 - From Show, select **Milestones Details** or **Summary**.
 - To filter the view, select a developer or release and click **Set**.
4. To email the report:
 - a. Click **Email**.
 - b. On the Email Report page, fill in the appropriate fields. Mandatory fields are marked with a red asterisk (*).
To learn more about an attribute, see field-level Help.
 - c. Click **Email**.
5. Click **Reset** to return the report to the default view.

Managing Feature Utilities

Use the Feature Utilities page to assign milestones to features, set feature due dates to milestone dates, or push due dates of open features.

To access Feature Utilities:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Feature Utilities**.
3. On the Feature Utilities, select a utility:
 - **Assign milestones to features (only those with no milestone currently assigned)**
 - **Set feature due date to milestone date (only affects features)**
 - **Push due date of open features**
4. Click **Next**.
5. Follow the on-screen instructions.

Renaming Focus Areas

When you create or update a feature, you can assign it to a focus area. Use Manage Focus Areas, to rename a focus area.

To rename an existing feature focus area:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Manage Focus Areas**.
The Manage Focus Areas page appears.
3. Locate the focus area and click the **Edit** icon.
4. In Rename To, enter your edits and click **Apply Changes**.

Downloading File Attachments

If enabled at the workspace-level, you can attach files to a feature, To Do, or bug. Use the Team Development Files page to download attached files.

Tip:

To use this feature, you must set the Enable File Repository attribute to **Yes** in Administration, Manage Service, Set Workspace Preferences. Administrators can also access this page in Team Development by clicking Enable Files on the Team Actions list.

To download files attached to features, To Dos, and bugs:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click the **View Files**.
The Team Development Files page appears.
3. To download a file, click the name.

See Also:

"Enabling Support for File Attachments in Team Development" in *Oracle Application Express Administration Guide*.

Purging Data

Remove data associated with a Team Development component type (that is, features, milestones, To Dos, bugs, or feedback).

To purge data associated with a component type:

1. On the Workspace home page, click the **Team Development** icon.

2. Under Utilities, click the **Purge Data**.
3. Under Purge Data, select the type of data to purge:
 - **Features**
 - **Milestones**
 - **To Dos**
 - **Bugs**
 - **Feedback**
4. Click **Purge**.

Managing News Entries

Use the News region to communicate with other users in the current workspace. News displays on the Application Express home page, the Team Development home page, and News page.

To add, edit, or delete new entries:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Manage News**.
The News page appears.
3. To add a news item:
 - a. Click **Add News**.
 - b. On the News page, enter text in the News Entry field.
 - c. Click **Add News**.
4. To edit a news item:
 - a. Select the news item.
 - b. Edit the News Entry.
 - c. Click **Apply Changes**.
5. To delete a news item:
 - a. Select the news item.
 - b. Click **Delete**.

Managing Links with Workspace Members

To add, edit, or delete links:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Manage Links**.
The News page appears.
3. To add a link:
 - a. Click **Create Link**.
 - b. On the Links page, edit the appropriate fields. Mandatory fields are marked with a red asterisk (*). To learn more about an attribute, see field-level Help.

- c. Click **Create Link**.
 - d. To access the link, click the Link Name.
 - 4. To edit a link:
 - a. Click the Edit icon adjacent to the Link Name.
 - b. On the Links page, edit the appropriate fields. Mandatory fields.
 - c. Click **Apply Changes**.
 - 5. To delete a link:
 - a. Click the Edit icon adjacent to the Link Name.
 - b. On the Links page, edit the appropriate fields. Mandatory fields.
 - c. Click **Delete**.

5

Creating Websheet Applications

Websheet applications enable end users to manage structured and unstructured data without developer assistance.

Websheet application pages contain sections, reports, and data grids linked together using navigation. Websheets can include navigation controls, search capabilities, and the ability to add annotations such as files, notes, and tags. You can secure Websheet applications using access control lists and several built-in authentication models.

- [Creating a Websheet Application](#)
Create a Websheet application in App Builder by running the Create Application Wizard, or by clicking Create in a running Websheet.
- [Copying an Existing Websheet Page](#)
Copy a Websheet application by selecting New Page as a Copy from the Create menu.
- [Running a Websheet](#)
To view a rendered version of your application, you run or submit it to the Application Express engine.
- [Editing Page Details](#)
Edit Page Details to change the page name, alias, parent page, or page description.
- [Understanding Websheet Properties](#)
Use the Websheet Properties page to control general application defaults and behavior.
- [Deleting a Websheet Application](#)
Delete a Websheet on the Websheet Properties page.
- [Viewing the Websheet Dashboard](#)
Use the Websheet dashboard to view details about the current Websheet application including recent changes, top users, and top pages.
- [Monitoring Websheet Activity](#)
Use the Monitor Activity page to analyze usage, performance, and popularity of pages and content.
- [Controlling Websheet Access](#)
Control access to a Websheet application by selecting an authentication mechanism (such as Oracle Application Express Account, Single Sign-On Verification, or LDAP Credentials Verification), or by creating a custom access control list.

 **See Also:**

- ["What is the Difference Between a Database and Websheet Application?"](#)
- ["Understanding AnyCo IT Department Websheet"](#)
- ["Using Websheets" in *Oracle Application Express End User Guide*](#)
- ["Removing and Validating Websheet Database Objects" and "Disabling the Creation of Sample Websheet Objects" in *Oracle Application Express Administration Guide*](#)

Creating a Websheet Application

Create a Websheet application in App Builder by running the Create Application Wizard, or by clicking Create in a running Websheet.

- [Creating a Websheet from App Builder](#)
- [Creating a Websheet Application from a Running Websheet](#)

 **See Also:**

- ["Using Websheets" in *Oracle Application Express End User Guide*](#)
- ["Removing and Validating Websheet Database Objects"](#)
- ["Disabling the Creation of Sample Websheet Objects"](#)
- [Oracle Application Express Administration Guide](#)

Creating a Websheet from App Builder

To create a Websheet application:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
The Create Application Wizard appears.
3. Select **Websheet** and click **Next**.
4. On Create Websheet:
 - a. **Websheet** - Enter a unique integer value to identify the application. Application IDs between 3000 to 9000 are reserved for internal use.
 - b. **Name** - Enter a unique integer value to identify the application.
 - c. **Include Getting Started Guide** - Select this option to include basic information about using Websheets in the first text section of the Websheet home page. Once reviewed, you can edit or remove this text section.

5. Confirm your selections and click **Create Websheet**.
A success message appears.
6. To view a rendered version of your application:
 - a. Click **Run Websheet**.
 - b. Enter your Websheet credentials:
 - i. Username - Enter your username.
 - ii. Password - Enter your password.
 - iii. Click **Log In** or **Sign In**.

 **Tip:**

The language that displays differs depending upon who creates the application. For Oracle-supplied applications, including productivity or sample applications and the Oracle Application Express development environment, **Sign In/Sign out** displays. For applications you or your developers create, **Log In/Log Out** displays.

The Websheet application appears in a new window.

Creating a Websheet Application from a Running Websheet

To create a Websheet application from a running Websheet:

1. Run a Websheet application.
2. From the Create menu, click **New Websheet**.
3. On Create Websheet:
 - a. Websheet - Enter a unique integer value to identify the application. Application IDs between 3000 to 9000 are reserved for internal use.
 - b. Name - Enter a name to identify the application.
 - c. Click **Next**.
4. Confirm your selections and click **Create** .
A success message appears.
5. To view a rendered version of your application:
 - a. Click **Run Websheet**.
 - b. Enter your Websheet credentials:
 - i. Username - Enter your username.
 - ii. Password - Enter your password.
 - iii. Click **Log In** or **Sign In**.

 **Tip:**

The language that displays differs depending upon who creates the application. For Oracle-supplied applications, including sample and productivity applications and the Oracle Application Express development environment, **Sign In/Sign out** displays. For applications you or your developers create, **Log In/Log Out** displays.

The Websheet application appears in a new window.

Copying an Existing Websheet Page

Copy a Websheet application by selecting New Page as a Copy from the Create menu.

To copy an existing Websheet page:

1. Run the Websheet application.
2. From the Create menu, select **New Page as a Copy**.
3. On New Page as a Copy, enter the name of the new page.
4. Click **Copy Page**.

A success message appears.

5. Select one of the following:
 - **View Current Page**
 - **View New Page**

 **Tip:**

You can change the parent page by editing the Page Details. See "[Editing Page Details](#)."

Running a Websheet

To view a rendered version of your application, you run or submit it to the Application Express engine.

- [Running a Websheet from the App Builder Home Page](#)
- [Running a Websheet from the Application Home Page](#)
- [Determining a Websheet URL](#)

 **See Also:**

- "Using Websheets" in *Oracle Application Express End User Guide*
- ["About Publishing the Websheet Application URL"](#)

Running a Websheet from the App Builder Home Page

To run a Websheet from the App Builder home page:

1. On the Workspace home page, click the **App Builder** icon.
2. From the Search bar, click the **View Report** icon.
3. Locate the Websheet application in the Applications list.
4. Click the **Run** icon in the far right column.
5. Enter your Websheet credentials:
 - a. Username - Enter your username.
 - b. Password - Enter your password.
 - c. Click **Log In** or **Sign In**.

 **Tip:**

The language that displays differs depending upon who creates the application. For Oracle-supplied applications, including sample and productivity applications and the Oracle Application Express development environment, **Sign In/Sign out** displays. For applications you or your developers create, **Log In/Log Out** displays.

The Websheet application appears in a new window.

Running a Websheet from the Application Home Page

To run an entire application from the Application home page:

1. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
2. Select the Websheet application.
The Websheet Properties page appears.
3. Click the **Run** button.
4. Enter your Websheet credentials:
 - a. Username - Enter your username.
 - b. Password - Enter your password.
 - c. Click **Log In** or **Sign In**.

 **Tip:**

The language that displays differs depending upon who creates the application. For Oracle-supplied applications, including sample and productivity applications and the Oracle Application Express development environment, **Sign In/Sign out** displays. For applications you or your developers create, **Log In/Log Out** displays.

The Websheet application appears in a new window.

Determining a Websheet URL

Once you complete your Websheet application, you can publish your production URL. To learn more, see "[About Publishing the Websheet Application URL](#)."

Editing Page Details

Edit Page Details to change the page name, alias, parent page, or page description.

To edit page details:

1. Run the Websheet application.
2. From the Edit menu, select **Edit Page**.
3. One Page Details.
 - Name - Identifies the name of the page. The page name displays in the window title and in the breadcrumb.
 - Alias - A page alias enables you to create links in page sections to other pages. A page alias must be unique within an application.
 - Owner - Identifies the owner of the page.
 - Page Description - Descriptive text that describes the page.
4. Click **Apply Changes**.

A success message appears.

 **Tip:**

You can also edit page details by clicking **Edit Page** on the Control Panel.

 **See Also:**

"[Running a Websheet](#)"

Understanding Websheet Properties

Use the Websheet Properties page to control general application defaults and behavior.

- [Editing Websheet Properties](#)
- [Websheet Properties](#)



See Also:

["Deleting a Websheet Application"](#)

Editing Websheet Properties

You can access the Websheet Properties page from inside or outside a running Websheet. The Home page identifies the default starting page users land on when running the Websheet.

To edit Websheet properties:

1. From the Application home page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page appears, select the Websheet application. The Websheet Properties appears.
2. From a running Websheet:
 - a. Run the Websheet.
 - b. From the Administration menu, select **Websheet Properties**. The Websheet Properties appears.
3. To show or hide a section, click the greater than sign (>) to the left of the section title.
4. Edit the appropriate Websheet properties as described in the next section.
5. Click **Apply Changes**.



See Also:

["Running a Websheet"](#)

Websheet Properties

Developers use the Websheet Properties page to control general application defaults and behavior. The Home Page attribute identifies the page, data grid, or report that users will land on by default.

- [Websheet](#)
- [Details](#)
- [Logo](#)
- [Style](#)
- [Authentication](#)
- [Authorization](#)
- [SQL](#)

Websheet

Websheet properties control general Websheet application behavior.

Table 5-1 Websheet Properties

| Attribute | Descriptions |
|-------------------------------|--|
| Name | Identifies the name of the application. This name displays within the breadcrumb when the application is running. |
| Status | Identifies the status of the application. |
| Home Page | Select the home page for this Websheet. This is the first page a user sees when running the Websheet. |
| Application Date Format | Determines the date format to be used in the application. This date format is used to alter the NLS_DATE_FORMAT database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If no value is specified, the default date format is derived from the database session at runtime. |
| Default Application Language | Identifies the default language of the application. Users are able to change their language preference from within the application. This language list is determined by the translated versions of Application Express which are installed. |
| Default Application Territory | Identifies the default territory of the application, which controls attributes like date language and numeric characters. Users are able to change their territory preference from within the application. |
| Show Reset Password | Controls the display of the Reset Password link on the Login page. This link enables users to reset their passwords. This option works only if the application is using Application Express Account authentication. |
| Websheet Email From Address | Determines the email address to use as the from address in the Websheet. This from email address is used when sending email from a data grid or report email download in the Websheet. |

Details

Details properties determine the application description and login message.

Table 5-2 Details

| Attribute | Descriptions |
|---------------|---|
| Owner | Identifies the user who owns the application. |
| Description | An optional description that describes the application. |
| Login Message | Display this message on the login page. |

Logo

Use Logo properties to create a logo for a Websheet application. This logo displays at the top of the Websheet.

Table 5-3 Logo

| Attribute | Descriptions |
|-----------|---|
| Logo Type | Select the type of logo being used for the application. |
| Logo | Enter a full text string to use as application logo. |

Style

Use the Style section to override default application style and use a custom CSS.

Authentication

Use Authentication properties to select an authentication scheme for this Websheet.

Table 5-4 Authentication

| Attribute | Descriptions |
|----------------|--|
| Authentication | <p>Select the way users of the application are authenticated. To change the authentication scheme, click Edit Authentication. The following list displays:</p> <ul style="list-style-type: none"> • Application Express Account - Authenticate using developer account username and password. • Single Sign-On - Authenticate using Oracle Application Server Single Sign-On. This must be configured for the Application Express instance. • LDAP - Authenticate using an LDAP directory. Specify additional parameters on this page to configure this. • Custom - Provide your own code for authentication and session management tasks. See "Controlling Websheet Access." |
| Logout URL | <p>Specify a URL to become this application's Logout URL attribute. This attribute is referenced in logout links as follows:</p> <pre>&LOGOUT_URL.</pre> <p>If the application uses built-in session verification, you may use "wwv_flow_custom_auth_std.ws_logout?p_websheet_app_id=&WS_APP_ID.&p_next_url=ws?p=&WS_APP_ID.:home" to invoke a built-in logout procedure.</p> |

Table 5-4 (Cont.) Authentication

| Attribute | Descriptions |
|---------------|--|
| Type | Type of session cookie used. You can use this attribute to share the same Application Express session between multiple applications in a workspace. With session sharing, you can seamlessly navigate between applications without logging in multiple times. <ul style="list-style-type: none"> • Application (No Sharing): The session cookie is specific to this application. Sessions are not shared with other applications. • Workspace Sharing: The session is shared with other applications of this workspace, if these also set Cookie Type to Workspace Sharing. • Custom: Define custom session cookie attributes. |
| Cookie Name | Name for the session cookie, which is required to identify an Application Express session and the session ID in the URL. If no value for the session cookie name is specified, Application Express picks a default value. |
| Cookie Path | Path attribute for the session cookie. |
| Cookie Domain | Domain attribute for the session cookie. |
| Secure | Options include: <ul style="list-style-type: none"> • Yes - Allow the session management cookie to be sent from the browser only when the protocol is HTTPS. • No - Allow the session management cookie to be sent from the browser when the protocol is HTTP or HTTPS. |

Authorization

Authorization properties control who can log in to a Websheet application.

Table 5-5 Authorization

| Attribute | Descriptions |
|--------------------------|--|
| Access Control List Type | If authentication is Application Express Account, choose whether authorization is determined by the role in the Application Express workspace, or if authorization is determined by the access control list. For all authentication types other than Application Express Account, authorization is always determined by the access control List. |
| Allow Public Access | Options include: <ul style="list-style-type: none"> • Yes - Enable users to run the application in a read-only mode without requiring users to log in. • No - All users must log in to run the application. |

 **Tip:**

Click **Access Control Roles** to learn more about roles and their privileges. Click **Edit Access Control List** to link to the Access Control List page. To learn more, see "[Controlling Websheet Access](#)."

SQL

SQL properties determine if users can use SQL in the current Websheet. By default, this capability is disabled. To change the default setting, select **Yes**.

If enabled, users can use SQL tags and SQL reports within the application. Click **Add Object** to link to the Suggestion Objects page. Users can write custom reports and use SQL tags on database objects within the schema associated with the current workspace. Identifying a database table or view as a Suggested Object makes it easier to write SQL by exposing the table and view columns in the help page.

 **Tip:**

Instance administrators can control the ability to use SQL for an entire instance. To learn more, see "Enabling and Disabling SQL Access in Websheets" in *Oracle Application Express Administration Guide*.

Deleting a Websheet Application

Delete a Websheet on the Websheet Properties page.

To delete a Websheet application:

1. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
2. Select the Websheet application.
The Websheet Properties page appears.
3. Click **Delete**.

 **See Also:**

"[Understanding Websheet Properties](#)"

Viewing the Websheet Dashboard

Use the Websheet dashboard to view details about the current Websheet application including recent changes, top users, and top pages.

To view the Websheet Dashboard:

1. Run the Websheet.
2. From the Administration menu, select **Dashboard**.
The Dashboard appears and is divided into the following sections:
 - Application Details
 - Recent Changes
 - Top Users
 - Top Pages
3. To edit the time frame, select a new value from the Timeframe list.
4. Click **Reset** to return to the default setting.



See Also:

["Running a Websheet"](#)

Monitoring Websheet Activity

Use the Monitor Activity page to analyze usage, performance, and popularity of pages and content.

To access the Monitor Activity page:

1. Log in to and run the Websheet.
2. From the Administration menu, select **Monitor Activity**.

The Monitor Activity page appears and is divided into the following sections:

- Activity
 - Page Views
 - Top Pages
 - Top Data Grids
 - Top Reports
 - Top Users
- Annotations
 - Top Files
 - Top Notes
 - Top Tags

Each report displays as an interactive report. To learn more about interactive reports and the Actions menu, see ["Customizing Interactive Reports in a Running Application."](#)

**See Also:**["Running a Websheet"](#)

Controlling Websheet Access

Control access to a Websheet application by selecting an authentication mechanism (such as Oracle Application Express Account, Single Sign-On Verification, or LDAP Credentials Verification), or by creating a custom access control list.

- [How Websheet Access Works](#)
- [Specifying the Access Control List Type](#)
- [Creating Access Control List Entries](#)

How Websheet Access Works

An administrator controls who can log in to a Websheet application by specifying one of the following authentication mechanisms:

- Public access (no authentication)
- Oracle Application Express Account
- Single Sign-On Verification
- LDAP Credentials Verification
- Custom

If authentication is defined as **Custom**, an access control list manages the privileges of this application's users.

If authentication is **Oracle Application Express Account**, you must specify whether authorization is determined by the user role defined in the Application Express workspace, or if authorization is determined by the access control list. For all authentication types other than Oracle Application Express Accounts, authorization is always determined by the access control list.

Once a user logs in to an application, a user's assigned role determines access. User assigned roles include:

- Reader - May view, but not edit content.
- Contributor - May view and edit content.
- Administrator - May view and edit content and edit Websheet Properties and manage the Access Control List.

Specifying the Access Control List Type

To use an access control list, you must edit the Websheet Properties and define the Access Control List Type as Custom.

To access the Websheet Properties page:

1. On the Workspace home page, click the **App Builder** icon.

2. On the App Builder home page appears, select the Websheet application.
The Websheet Properties appears.
3. Under Authorization:
 - a. Access Control List Type - Select **Custom**.
 - b. Allow Public Access - When set to **Yes** to enable users to run the application in a read-only mode without requiring users to log in. When set to **No**, all users need to log in to run the application.
4. Click **Apply Changes**.



See Also:

["Authentication"](#)

Creating Access Control List Entries

To create access control list entries:

1. Run the Websheet.
2. From the Administration menu, select **Access Control**.
3. Click **Create Entry**.
The Entry Details page appears.
4. On the Entry Details page:
 - a. Username - Identify the username that you wish to control access for.
 - b. Privilege - Identify the privilege level for the user. If the application is using custom authorization, the access control list manages the privileges of this application's users. Option include:
 - **Reader** - May view, but not edit content.
 - **Contributor** - May view and edit content.
 - **Administrator** - May view and edit content and edit Websheet Properties and manage the Access Control List.
5. Click either **Create** or **Create and Create Another**.



See Also:

["Running a Websheet"](#)

6

Creating Database Applications

A database application is a collection of pages linked together using navigation menus, tabs, buttons, or hypertext links. Application pages share a common session state and authentication.

To create a database application, a developer runs wizards to declaratively assemble pages and navigation. Individual pages are organized using containers called **regions**. Once created, developers can modify an application by editing application attributes and add new pages using the Create Page Wizard. Developers can further manage the development process by creating page groups, locking and unlocking pages, adding developer and application comments, and accessing a variety of application, page, and cross page utilities and reports.

- [About the Create Application Wizard](#)
To create an Oracle Application Express application, a developer runs the Create Application Wizard.
- [Understanding Page Types, Features, and Settings](#)
Use the Create Application Wizard to build a complete application containing multiple pages. How the Create Application Wizard works depends upon the type of pages you are creating.
- [Creating a Database Application Based on a Table or Query](#)
Select **New Application** in the Create Application Wizard to create a database application containing multiple pages based on a table you select, or by providing a valid SQL query.
- [Creating a Database Application from a Spreadsheet](#)
Select **From a Spreadsheet** in the Create Application Wizard to load spreadsheet data and then create report on that data.
- [Using Application Blueprints](#)
View an application blueprint and directly edit the application definition in JSON format.
- [Copying a Database Application](#)
Create a copy of an existing application.
- [Creating Applications for Mobile Devices](#)
Build applications for mobile devices with mobile-specific design patterns that can provide an optimal user experience for small screens.
- [Deleting an Application](#)
You can delete an application from the Application home page, or while editing application attributes.
- [Managing Application Attributes](#)
Application attributes (also known as the Application Definition) control the behavior of an entire application and are divided into the categories: Definition, Security, Globalization, and User Interface.

- [Adding Developer Comments](#)
Add comments to an application, a page, or a group of pages using the Developer Comment, Bug, or To Do button. You can use developer comments to communicate application changes, report issues, or record developer suggestions.
- [Using the Find Icon](#)
The Find icon resembles a flashlight and displays on many pages in App Builder including Application home page, Shared Components, and the Edit Application Definition. Click the Find icon to search for items, pages, queries, tables, PL/SQL code, or images, view debug reports, view session state, and view errors.
- [Adding Database Application Comments](#)
Use Application Comments to describe an application or track developers involved in the application development.
- [Managing Application Groups](#)
Organize applications by assigning them to application groups. To use application groups, first create a group and then assign applications to it.
- [Using Application Utilities](#)
Access the Utilities page to view useful application summaries and reports.



See Also:

["What is the Difference Between a Database and Websheet Application?"](#)

About the Create Application Wizard

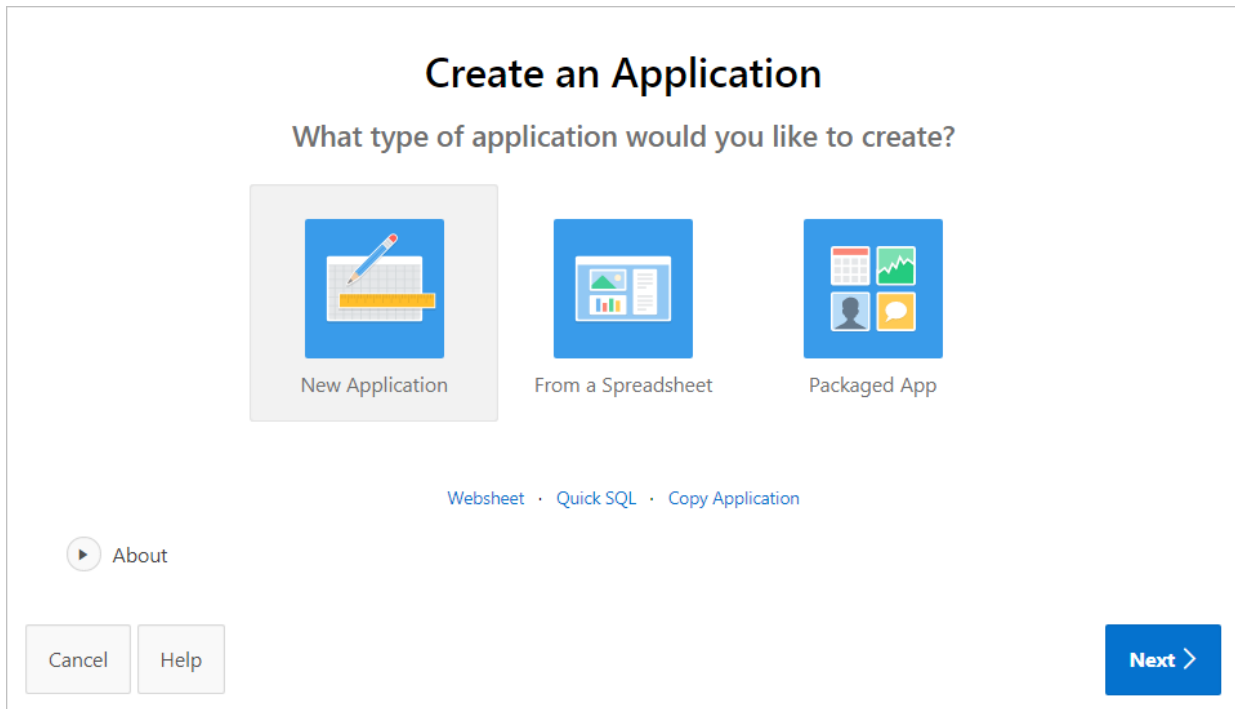
To create an Oracle Application Express application, a developer runs the Create Application Wizard.

To run the Create Application Wizard, click the **Create** button on the App Builder home page.



Tip:

The first time you access the App Builder home page, grey box labeled **Create a New App** appears. Clicking **Create New App** is a shortcut to accessing the Create Application Wizard.



When you run the Create Application Wizard, the wizard prompts you to choose the type of application you want to create. Options include:

- **New Application.** Create a fully functional database application based on tables you select or by providing a valid SQL. You can add pages that include various components including calendars, cards, charts, dashboards, forms, interactive grids, master detail or editable grids, and reports. Add application-level features such as an Application About page, role-based user authentication, end user activity reports, configuration options to enable or disable specific functionality, a feedback mechanism to gather end users comment, and a Customize button to enable end users to choose their own theme style
- **From a Spreadsheet.** Load spreadsheet data from a spreadsheet by importing it from a file as comma separated (*.csv) or tab delimited file, or by copying and pasting tab delimited data. Once you approve a page preview, the wizard loads the data into the database, creates a report and form on the data, and then displays the Create Application Wizard to complete the application creation process.
- **App Gallery.** Links to the App Gallery. The App Gallery include a set of business productivity and sample applications which can be installed with just a few clicks. **Productivity apps** are fully developed point-solutions designed to provide real functionality, such as project management, surveys, shared calendars, and tracking applications. Productivity applications can be installed, run and removed. By default they are 'locked' and are fully supported. Once unlocked, the application is no longer supported but it can be updated to meet specific requirements. **Sample apps** are not a complete applications but instead contains code snippets or sample code. Sample applications are available for installation in a workspace and are editable by default.
- **Websheet.** Websheet applications enable users to build data centric applications without any SQL programming knowledge. Websheet applications are simplified

applications that support pages, data grids, and reports. By creating These applications are very easy to build and are designed to support community contributions.

- **Quick SQL.** Generate the SQL required to create a relational data model from an indented text document. Quick SQL is designed to reduce the time and effort required to create SQL tables, triggers, and index structures.
- **Copy Application.** Create a copy of an existing application.

See Also:

- ["Creating a Database Application Based on a Table or Query"](#)
- ["Creating a Database Application from a Spreadsheet"](#)
- ["Utilizing the App Gallery"](#)
- ["Creating Websheet Applications"](#)
- ["Using Quick SQL" in *Oracle Application Express SQL Workshop Guide*](#)
- ["Copying a Database Application"](#)

Understanding Page Types, Features, and Settings

Use the Create Application Wizard to build a complete application containing multiple pages. How the Create Application Wizard works depends upon the type of pages your are creating.

- [About Creating a New Database Application](#)
Use the Create Application Wizard to design and quickly create Oracle Application Express applications.
- [Available Page Types in the Create Application Wizard](#)
Learn about available page types when running the Create Application Wizard.
- [Available Features in the Create Application Wizard](#)
Learn about available features when running the Create Application Wizard. Features provide application-level functionality and can only be added once per application.
- [Available Settings in the Create Application Wizard](#)
Learn about Settings available when running the Create Application Wizard. Settings are used in the generation of the application and include the application ID, the database schema, Advanced Settings (such as the application definition, security, and globalization attributes), and application authentication.
- [About Removing Features Created with the Create Application Wizard](#)
Remove features created with a wizard using build options.

 **See Also:**

- ["Creating a Database Application Based on a Table or Query"](#)
- ["Creating Applications for Mobile Devices"](#)

About Creating a New Database Application

Use the Create Application Wizard to design and quickly create Oracle Application Express applications.

Creating a new database application with the Create Application Wizard is a multiple step process. Once you specify the application name and appearance, you add pages. Database applications can contain multiple pages that include various components such as calendars, cards, charts, dashboards, forms, interactive grids, master detail or editable grids, and reports. Once created, you can edit the page names, alter the page order, and delete them.

Features provide application-level functionality and can only be added once per application. Available features include an Application About page, role-based user authentication, end user activity reports, configuration options to enable or disable specific functionality, a feedback mechanism to gather end users comment, and a Customize button to enable end users to choose their own theme style.

Then, you configure **Settings**. Settings are used in the generation of the application and include the application ID, the database schema, Advanced Settings (such as the application definition, security, and globalization attributes), and application authentication.

After you create an application using the Create Application Wizard, you can modify pages and add additional pages using the Create Page Wizard.

 **See Also:**

- ["Managing Pages in a Database Application"](#)

Available Page Types in the Create Application Wizard

Learn about available page types when running the Create Application Wizard.

Available Page Types

Table 6-1 Create Application Wizard - Available Page Types

| Page Type | Description | To Learn More |
|-----------|---|--|
| Blank | Create a blank page as a placeholder. Once you create the application, you can create regions on the page in Page Designer. | See online Help when creating this page. |

Table 6-1 (Cont.) Create Application Wizard - Available Page Types

| Page Type | Description | To Learn More |
|------------------|---|---|
| Calendar | Generates a calendar with monthly, weekly, and daily views. | See " Creating Calendars " |
| Cards | Create a page which displays cards. | See online Help when creating this page. |
| Chart | Create a page which displays either an area, bar, line, or pie chart. | See " Creating Charts " |
| Dashboard | Create a dashboard page with multiple charts. | See online Help when creating this page. |
| Form | Create a page containing a form which enables end users to maintain data. Select the table on which to build the form and specify whether or not to include a report. | See " Developing Forms " |
| Interactive Grid | <p>Create a page which displays as an interactive grid (similar to a spreadsheet). Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse.</p> <p>You choose a table or view on which to build the interactive grid, or provide valid SQL statement which returns distinct columns.</p> <p>You also specify whether or not it is editable. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.</p> | See " Managing Interactive Grids " |
| Master Detail | <p>Create a master detail form that enables users to query, insert, update, and delete values from two related tables or views. You choose the tables on which to build the master and detail regions. Master Detail options include:</p> <ul style="list-style-type: none"> • Side by Side - Create a single page master detail utilizing side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report. • Stacked - Creates a single page master detail with editable interactive grids. Users select a row in the master grid to update the detail grids. | See " Managing Master Detail Forms " |
| Report | <p>Creates a page that contains the formatted result of a SQL query. You choose a table on which to build a report, or provide a custom SQL SELECT statement or a PL/SQL function returning a SQL SELECT statement. You then choose the report type (that is, (that is, Interactive Report or Classic Report).</p> <p>Select Include Form to include a form page for creating or updating records.</p> | See " Creating an Interactive or Classic Report Using the Create Application Wizard " |
| Timeline | Create a page which displays a timeline. Especially useful for displaying updates. | See online Help when creating this page. |
| Wizard | Create a collection of pages to be used as a wizard. Generally wizards are used for entering data across multiple steps. | See online Help when creating this page. |

Table 6-1 (Cont.) Create Application Wizard - Available Page Types

| Page Type | Description | To Learn More |
|------------------|---|-----------------|
| Multiple Reports | Create multiple interactive reports based on the tables you select. | Not applicable. |

Available Features in the Create Application Wizard

Learn about available features when running the Create Application Wizard. Features provide application-level functionality and can only be added once per application.

This table lists available Features when running the Create Application Wizard.

Table 6-2 Create Application Wizard — Features

| Page Type | Description | To Learn More |
|-----------------------|--|--|
| About Page | Include an About page in the application which displays the application description. | See online Help when creating feature. |
| Access Control | Incorporate role based user authentication within your application. Users can be defined as Administrators, Contributors, or Readers. You can then readily define different access to different roles for various components throughout your application, such as pages, menu entries, regions, columns, items, buttons and so forth. | See " Controlling Access to Applications, Pages, and Page Components " |
| Activity Reporting | <p>Include numerous reports on end user activity for your application. Determine the most active users, the most used pages, the performance of pages, and errors raised, to better understand how your application is being utilized and areas for improvement.</p> <ul style="list-style-type: none"> • Top Users report • Application Error Log report • Page Performance, activity and performance by page • Application activity by page report • Page Views detail report | Not applicable. |
| Configuration Options | Enables application administrators to enable or disable specific functionality within the application. This feature is useful if you select features that need additional development effort before they can be used by end users. This feature can also be expanded to application-specific features. If developers define additional build options and associate them with specific functionality throughout the application, then they can be added to the configuration settings for administrators. For example, within the <i>Customer Tracker</i> productivity app, administrators can turn on or off such features as Contacts, Data Loading, Geography, and more. | Not applicable. |

Table 6-2 (Cont.) Create Application Wizard — Features

| Page Type | Description | To Learn More |
|-----------------------|--|---|
| Feedback | <p>Feedback provides a mechanism for end users to post general comments for application administrators and developers. The posts include useful session state information to help developers determine where the end user sent the feedback from.</p> <p>Creating Feedback:</p> <ul style="list-style-type: none"> • Creates a Navigation bar icon which users can click to leave feedback. • Creates a report for viewing and updating feedback. • Captures the application and page ID, feedback comments, date and time, and user information. | See " Managing Feedback " |
| Theme Style Selection | <p>Enables administrators to select a default color scheme (theme style) for the application. Administrators determine whether end users can choose their own theme style by enabling and disabling Allow End Users to choose Theme Style is enabled.</p> <p>If enabled, end users simply click on the Customize link at the bottom of the home page and select from the available theme styles. For example, users with visual impairment may prefer to utilize the Vista theme style which has a much higher color contrast</p> | Not applicable. |



See Also:

["About Removing Features Created with the Create Application Wizard"](#)

Available Settings in the Create Application Wizard

Learn about Settings available when running the Create Application Wizard. Settings are used in the generation of the application and include the application ID, the database schema, Advanced Settings (such as the application definition, security, and globalization attributes), and application authentication.

This table lists available Settings when running the Create Application Wizard.

Table 6-3 Create Application Wizard — Settings

| Page Type | Description | To Learn More |
|----------------|--|-----------------|
| Application ID | Unique, numeric identifier for your application. This field contains an automatically generated identifier by default. | Not applicable. |

Table 6-3 (Cont.) Create Application Wizard — Settings

| Page Type | Description | To Learn More |
|----------------|---|---|
| Schema | Select the database schema which stores the database objects you want to use in this application. | Not applicable. |
| Language | Select the primary language for this application. | Not applicable. |
| Authentication | Select how you want users to authenticate into your application. | See " Establishing User Identity Through Authentication " |

Table 6-3 (Cont.) Create Application Wizard — Settings

| Page Type | Description | To Learn More |
|-------------------|--|-----------------|
| Advanced Settings | <p>Set additional definition, security, and globalization settings. These settings can be applied when creating the application and can be edited post application creation.</p> <p>Tip: To learn more about these attributes, see field-level Help.</p> <p>User Interface Attributes:</p> <ul style="list-style-type: none"> • Apply User Interface Defaults - Select Yes to have attribute defaults based on existing User Interface Defaults defined within this workspace. • Copy from Existing Apps - Select Yes to have attribute defaults to be based on existing Form pages, based on the same table, in current applications defined within this workspace. • Table Prefixes - Enter table prefixes currently used by existing database tables on which pages are based. These prefixes may be different from the framework table prefix (for database objects created by this application when generated). <p>Description:</p> <ul style="list-style-type: none"> • Short Description - Enter text to be displayed as a subtitle, under the application name on the home page. • Description - Enter text to be displayed on the About This Application page, accessed from the help icon in the navigation bar. <p>Settings:</p> <ul style="list-style-type: none"> • Add "Built with APEX" to Footer - When set to Yes, Oracle Application Express adds the text "Built with Love using Oracle APEX" to the footer of every page. • Version - Enter the application version. The version is displayed by default on the bottom of each page. • Logging - Determines whether or not user activity is recorded in the activity log. When set to Yes, every page view is logged, enabling an administrator to monitor user activity for each application. • Debugging - Determines whether debug mode can be enabled using the browser. <p>Security:</p> <ul style="list-style-type: none"> • Deep Linking - Enables or prevents deep linking to an application. • Maximum session length in seconds - Defines how long (in seconds) sessions can exist and be used by this application. | Not applicable. |

Table 6-3 (Cont.) Create Application Wizard — Settings

| Page Type | Description | To Learn More |
|-------------------------|---|---|
| | <ul style="list-style-type: none"> • Maximum session idle time in seconds - Defines the time between the last page request and the next page request. Globalization: <ul style="list-style-type: none"> • Document Direction - Set document direction left-to-right or right-to-left. • Date Format - Determines the date format to be used in the application. • Date Time Format - Specify the date time format to be used in the application. • Timestamp Format - Determines the timestamp format to be used in the application. • Timestamp with Timezone Format - Determines the timestamp with time zone format to be used in the application. | |
| User Interface Defaults | Set defaults for user interface settings, such as rows per page, and add list of values definitions. These settings are used when generating components within the application. | See "Oracle Application Express SQL Workshop Guide" |

About Removing Features Created with the Create Application Wizard

Remove features created with a wizard using build options.

When you add a feature using a wizard, the wizard creates one or more pages and other components and processes to seamlessly integrate it into your application. Each feature is associated with a build option which contains one or more components. You use the associated build option to enable, disable, or permanently remove features.

Using Build Options to Include or Exclude Features

Build options have two possible values: **Include** and **Exclude**. If you select the build option status of **Include**, then the Application Express engine considers the associated components (in this case features) as part of the application definition at runtime. Conversely, if you specify the build options status as **Exclude**, then the Application Express engine treats it and any associated components as if it did not exist.

Removing Features Permanently

You remove features by first removing the components associated with the build option and then deleting the build option.

See Also:

- ["Including or Excluding Build Options"](#)
- ["Deleting Build Options and Associated Components"](#)

Creating a Database Application Based on a Table or Query

Select **New Application** in the Create Application Wizard to create a database application containing multiple pages based on a table you select, or by providing a valid SQL query.

To create a database application:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
The Create Application Wizard appears.
3. Click **New Application**.
The Create an Application page appears.
4. For Name, enter the name used to identify the application to developers.
5. For Appearance, you can either:
 - Accept the default (**Vita, Side Menu**).
 - Change the appearance by clicking the **Set Appearance** icon adjacent to the Appearance field. The Appearance dialog appears.
 - Theme Style - Select the default theme style used to display the application.

 **Tip:**

You can change the style to another pre-defined style, or modify it post generation. To change the style run the application and then click **Theme Roller** in the Developer toolbar.

- Navigation - Select if the menu displays on the side of the application or across the top of the application.
 - Application Icon - Click **Choose New Icon** to select the icon color and one of the available icons. Your selection displays on the home page of the application.
 - Click **Save Changes**.
6. To add a page, click **Add Page** and select the desired page type. The user interface changes based on the selected page type.

Pages you create display under the Home page.

You can edit existing pages as follows:

- **Change the page order.** To change the order in which pages appear in your application, click and hold the **Drag to reorder page** icon and drag and drop it to a new location in the list.

The Home page always displays first and cannot be reordered. Administrative pages always display at the bottom of the list and the order dictates the order they appear in the Application Administration list on the Administration page.

- **Edit a page.** To edit a page click **Edit**. In the dialog, edit the page name, change the icon, specify if the page is a Home Page or Administration Page, or define Page Help.
 - **Delete a page.** To delete a page, click **Edit** and the click **Delete** .
7. For Features, select features to include with your application. Features provide application-level functionality and can only be added once per application. To learn more, see Help.

 **Tip:**

Click the **Check All** button to select all features.

8. For Settings:
- a. Application ID - Enter a unique, numeric identifier for your application. This field contains an automatically generated identifier by default. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - b. Schema - Select the database schema which stores the database objects you want to use in this application. Each application obtains its privileges by parsing all SQL as a specific database schema.
 - c. Languages - The primary language used in the app. To change the primary language or translate the app into additional languages, click the **Select Languages** icon.
 - d. Authentication - Select how you want users to authenticate into your application..
 - e. Advanced Settings - Click the icon adjacent to the Advanced Settings to edit the application definition settings and preferences and security and globalization attributes. To learn more, see Help.
 - f. User Interface Defaults - Click the icon to apply User Interface Defaults to this application.
9. Click **Create Application**.

 **See Also:**

- ["Available Page Types in the Create Application Wizard"](#)
- ["Available Features in the Create Application Wizard"](#)
- ["Available Settings in the Create Application Wizard"](#)

Creating a Database Application from a Spreadsheet

Select **From a Spreadsheet** in the Create Application Wizard to load spreadsheet data and then create report on that data.

- [About Creating an Application from a Spreadsheet](#)
Load spreadsheet data from a file or by copying and pasting and then create an interactive report on that data.
- [Loading Spreadsheet Data from a File](#)
Load spreadsheet data from a file as comma separated (*.csv) or tab delimited file and then create an interactive report on that data.
- [Loading Spreadsheet Data by Copying and Pasting](#)
Select From a Spreadsheet in the Create Application Wizard to load spreadsheet data and then create report on that data.

About Creating an Application from a Spreadsheet

Load spreadsheet data from a file or by copying and pasting and then create an interactive report on that data.

When you run the Create Application Wizard and select **From a Spreadsheet**, the Create From a Spreadsheet Wizard appears.

You then choose how to upload the spreadsheet data. Options include:

- **From a file.** Select this option to upload spreadsheet data from a file as comma separated (*.csv) or tab delimited file.
- **Copy and paste.** Select this option to copy and paste tab delimited data.

Once you approve a page preview, the wizard loads the data into the database, creates a report and form on the data, and then displays the Create Application Wizard. A new interactive report with form page named after the table you imported displays under **Pages**. You then finish the application creation process by naming the application, specifying the application appearance, editing the report, create additional pages, selecting features, and configuring settings.

Copy and Paste Sample Data

If you choose the **Copy and paste** option, the wizard includes Sample Data. Simply expand the **Sample Data** region and select a sample.

Support for Uploading JSON Data

The Create From a Spreadsheet Wizard supports the upload of JSON data. To view two examples, select **Copy and paste** option, expand the **Sample Data** region, and select either **JSON and Structured Test Data** or **JSON only Test Data**.

Loading Spreadsheet Data from a File

Load spreadsheet data from a file as comma separated (*.csv) or tab delimited file and then create an interactive report on that data.

To load spreadsheet data from a file:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
The Create Application Wizard appears.
3. Select **From a Spreadsheet**.
4. When prompted how to load your spreadsheet data, select **From a File**.

You can upload a comma separated (*.csv) or tab delimited file. The file you upload must be text-based. To upload a *.xls file, you must first save it as *.csv.

5. On From a Spreadsheet:
 - a. Text File - Click **Choose File** to locate the file to be uploaded.
 - b. Separator - Specify the column separator character. Use \t for tab separators.
 - c. Optionally Enclosed By - Enter a delimiter character. You can use this character to delimitate the starting and ending boundary of a data value. If you specify a delimiter character, the wizard ignores white space occurring before the starting and ending boundary of a data value. You can also use this option to enclose a data value with the specified delimiter character.
 - d. First row contains column names - If you enable this option and your data contains column names in the first row, the wizard will use the column names for the Oracle database table. Otherwise, you are prompted to update the column names manually.
 - e. File Character Set - Choose the character set in which the text file is encoded.
 - f. Globalization - Expand the region and specify the Currency Symbol, Group Separator, and Decimal Character..

To learn more about an attribute, see field-level Help.
 - g. Click **Next**.

- 6.
7. On Page Preview:
 - a. Schema - Identify the database schema owner. Your application obtains its privileges by parsing all SQL as a specific database schema.
 - b. Table Name - Enter a name for the table being created.
 - c. Preserve Case - If applicable, select this option to preserve the existing case.
 - d. Set Table Properties - Review the table properties. The wizard determines the data types and column length for each column by reviewing the data contained in the first 20 rows of data.
 - Modify the data types or enter format masks.
 - Specify whether to include a column by selecting **Yes** or **No** from the Upload list.
8. Click **Load Spreadsheet**.
9. Click **Continue to Create Application Wizard**.

The Create Application Wizard appears. A new **Interactive Report with Form** page named after the table you created displays under **Pages**. From here you can edit the report (such as change the report name) or create additional pages based on the imported data.

 **Tip:**

The steps that follow summarize how to use the Create Application Wizard. For more information, see field-level help.

10. For Name, enter the name used to identify the application to developers.

11. For Appearance, you can either Accept the default (**Vita, Side Menu**), or change the appearance by clicking the **Set Appearance** icon and edit the attributes in the Appearance dialog.
12. To add a page, click **Add Page** and select the desired page type. The user interface changes based on the selected page type.

Pages you create display under Add Page. You can edit existing pages as follows:

- **Change the page order.** To change the order in which pages appear in your application, click and hold the **Drag to reorder page** icon and drag and drop it to a new location in the list.

The Home page always displays first and cannot be reordered. Administrative pages always display at the bottom of the list and the order dictates the order they appear in the Application Administration list on the Administration page.
 - **Edit a page.** To edit a page click **Edit**. In the dialog, edit the page name, change the icon, specify if the page is a Home Page or Administration Page, or define Page Help.
 - **Delete a page.** To delete a page, click **Edit** and then click **Delete**.
13. For Features, select features to include with your application. Features provide application-level functionality and can only be added once per application. To learn more, see Help.

 **Tip:**

Click the **Check All** button to select all features.

14. For Settings:
 - a. Application ID - Enter a unique, numeric identifier for your application. This field contains an automatically generated identifier by default. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - b. Schema - Select the database schema which stores the database objects you want to use in this application. Each application obtains its privileges by parsing all SQL as a specific database schema.
 - c. Application Languages - The primary language used in the app. To change the primary language or translate the app into additional languages, click the **Select Languages** icon.
 - d. Advanced Settings - Click the icon adjacent to the Advanced Settings to edit the application definition settings and preferences and security and globalization attributes. To learn more, see Help.
 - e. Authentication - Select how authenticate users.
 - f. User Interface Defaults - Click the icon to apply User Interface Defaults to this application.
15. Click **Create Application**.

 **See Also:**

- ["Available Page Types in the Create Application Wizard"](#)
- ["Available Features in the Create Application Wizard"](#)
- ["Available Settings in the Create Application Wizard"](#)

Loading Spreadsheet Data by Copying and Pasting

Select **From a Spreadsheet** in the Create Application Wizard to load spreadsheet data and then create report on that data.

To load spreadsheet data by copying and pasting:

1. On the Workspace home page, click the **App Builder** icon.

2. Click the **Create** button.

The Create Application Wizard appears.

3. Select **From a Spreadsheet**.

4. When prompted how to load your spreadsheet data, select **Copy and paste**.

You can upload a comma separated (*.csv) or tab delimited file. The file you upload must be text-based. To upload a *.xls file, you must first save it as *.csv.

5. On **From a Spreadsheet**:

a. **Copy and Paste Tab Delimited Data** - Copy and paste the tab delimited data you want to import.

b. **First row contains column names** - If you enable this option and your data contains column names in the first row, the wizard will use the column names for the Oracle database table. Otherwise, you are prompted to update the column names manually.

c. **File Character Set** - Choose the character set in which the text file is encoded.

d. **Globalization** - Expand the region and specify the Currency Symbol, Group Separator, and Decimal Character. To learn more, see field-level Help.

To learn more about an attribute, see field-level Help.

e. **Sample Data** - Expand the region and select a sample. Once selected, the sample appears in **Copy and Paste Tab Delimited Data**.

f. Click **Next**.

6.

7. On **Page Preview**:

a. **Schema** - Identify the database schema owner. Your application obtains its privileges by parsing all SQL as a specific database schema.

b. **Table Name** - Enter a name for the table being created.

c. **Preserve Case** - If applicable, select this option to preserve the existing case.

- d. Set Table Properties - Review the table properties. The wizard determines the data types and column length for each column by reviewing the data contained in the first 20 rows of data.
 - Modify the data types or enter format masks.
 - Specify whether to include a column by selecting **Yes** or **No** from the Upload list.
8. Click **Load Spreadsheet**.
9. Click **Continue to Create Application Wizard**.

The Create Application Wizard appears. A new **Interactive Report with Form** page named after the table you created displays under **Pages**. From here you can edit the report (such as change the report name) or create additional pages based on the imported data.

 **Tip:**

The steps that follow summarize how to use the Create Application Wizard. For more information, see field-level help.

10. For Name, enter the name used to identify the application to developers.
11. For Appearance, you can either Accept the default (**Vita, Side Menu**), or change the appearance by clicking the **Set Appearance** icon and edit the attributes in the Appearance dialog.
12. To add a page, click **Add Page** and select the desired page type. The user interface changes based on the selected page type.

Pages you create display under Add Page. You can edit existing pages as follows:

- **Change the page order.** To change the order in which pages appear in your application, click and hold the **Drag to reorder page** icon and drag and drop it to a new location in the list.

The Home page always displays first and cannot be reordered. Administrative pages always display at the bottom of the list and the order dictates the order they appear in the Application Administration list on the Administration page.
 - **Edit a page.** To edit a page click **Edit**. In the dialog, edit the page name, change the icon, specify if the page is a Home Page or Administration Page, or define Page Help.
 - **Delete a page.** To delete a page, click **Edit** and the click **Delete** .
13. For Features, select features to include with your application. Features provide application-level functionality and can only be added once per application. To learn more, see Help.

 **Tip:**

Click the **Check All** button to select all features.

14. For Settings:

- a. Application ID - Enter a unique, numeric identifier for your application. This field contains an automatically generated identifier by default. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - b. Schema - Select the database schema which stores the database objects you want to use in this application. Each application obtains its privileges by parsing all SQL as a specific database schema.
 - c. Application Languages - The primary language used in the app. To change the primary language or translate the app into additional languages, click the **Select Languages** icon.
 - d. Advanced Settings - Click the icon adjacent to the Advanced Settings to edit the application definition settings and preferences and security and globalization attributes. To learn more, see Help.
 - e. Authentication - Select how authenticate users.
 - f. User Interface Defaults - Click the icon to apply User Interface Defaults to this application.
15. Click **Create Application**.

Using Application Blueprints

View an application blueprint and directly edit the application definition in JSON format.

- [What Is an Application Blueprint?](#)
- [Viewing an Application Blueprint](#)
- [Loading an Application Blueprint](#)

See Also:

To learn more about JSON, see <https://www.json.org/>

What Is an Application Blueprint?

You can view the application blueprint by running the Create Application Wizard and clicking the **View Blueprint** link at the top of the Create Application Wizard. The Application Blueprint window transforms and displays the current application definition as a JSON document. Edit the document to update application properties and then click **Apply Blueprint** to change the current application details.

If you need to create a large number of similar pages you can copy the relevant JSON code for one page, paste it numerous times, and then make small adjustments to the JSON. This approach can be significantly quicker than manually clicking the **Add Page** button for each page in the wizard.

You can also replace the blueprint with the definition from a previously generated application. Click **Load a Previous Blueprint** to replace the blueprint with the definition from a previously generated application. This approach is a simple way to

iterate the application design by adding additional pages, changing the features, or tweaking the settings, before generating the application again.

Viewing an Application Blueprint

To view an application blueprint:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Click **New Application**.
4. At the top of the page, click the **View Blueprint** link.

The Application Blueprint window appears displaying the application definition in JSON format.

5. To edit the document:

- a. Update application definition.

The following buttons display at the top of the Script Editor:

- **Undo - Ctrl+Z** — Removes, or undoes, the most recent line edit made in the Script Editor.
- **Redo - Ctrl+Y** — Repeats the most recent line edit made in the Script Editor.
- **Find - Ctrl+F** — **Find** resembles a magnifying glass. Click **Find** to perform a basic search.
- **Replace - Ctrl+Shift+F** — **Replace** resembles a two-sided arrow. Click **Replace** to search and replace existing code.

- b. Click **Apply Blueprint** to change the current application definition.

6. To replace the blueprint with the definition from a previously generated application:

- a. Click **Load Previous Blueprint**.
- b. In Load Blueprint, locate the blueprint and click **Select**.
- c. Click **Apply Blueprint** to change the current application details.

Loading an Application Blueprint

To view an application blueprint:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Click **New Application**.
4. At the top of the page, click the **Load Blueprint** link.

The Load Blueprint window appears.

5. On Load Blueprint, you can:

- **Load a new blueprint.** Locate the blueprint to load and click **Select**.
- **Delete a blueprint.** Click the **Delete Blue** icon.

Copying a Database Application

Create a copy of an existing application.

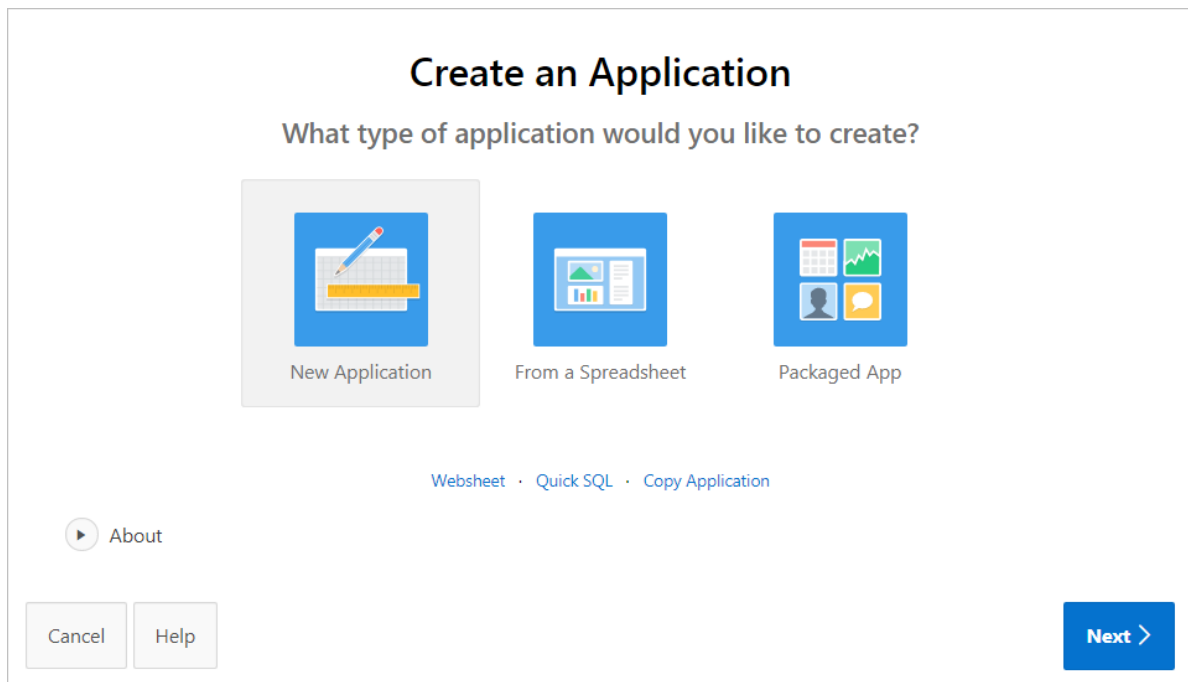
Create a copy of an existing application by running the Create Application Wizard and selecting **Copy an existing application**, or by selecting the application and then **Copy this Application** from the Tasks list on the Application home page.

- [Copying a Database Application Using the Create Application Wizard](#)
- [Copying an Application from the Application Home Page](#)

Copying a Database Application Using the Create Application Wizard

To copy application by running the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Select **Copy Application** and click **Next**.



4. For Identify Copy Target:
 - a. Copy From - Select the name of the application to copy.
 - b. Copy To - Enter a short descriptive name for the application to distinguish the new application in select lists and reports.
 - c. New Application ID - Enter a unique integer value to identify the application. Application IDs between 3000 to 9000 are reserved for internal use.
 - d. Copy Supporting Object Definitions - Specify whether to include supporting object definitions. Options include:

- **Yes** - Include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file.
 - **No** - Do not include supporting object definitions.
- e. Click **Next**.
5. Click **Create Application**.

Copying an Application from the Application Home Page

To copy a database application from the Application home page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application to copy.
The Application home page appears.
3. From the Tasks list, click **Copy this Application**.
4. For Identify Copy Target:
 - a. New Application ID - Enter a unique integer value to identify the application. Application IDs between 3000 to 9000 are reserved for internal use.
 - b. New Application Name - Enter a short descriptive name for the application to distinguish the new application in select lists and reports.
 - c. Copy Supporting Object Definitions - Specify whether to include supporting object definitions. Options include:
 - **Yes** - Include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file.
 - **No** - Do not include supporting object definitions.
 - d. Click **Next**.
5. Click **Copy Application**.

Creating Applications for Mobile Devices

Build applications for mobile devices with mobile-specific design patterns that can provide an optimal user experience for small screens.

- [About Mobile Application Design](#)
- [About Mobile Support](#)
- [Creating Mobile Applications](#)

About Mobile Application Design

Mobile applications developed with Oracle Application Express are browser-based applications that run inside the browser on the mobile device. Therefore, these applications must have a connection in order to communicate with the Oracle Database and cannot operate in a disconnected environment.

Mobile devices that have HTML5 capabilities can utilize all of the capabilities that can be built into the applications, including HTML5 date-pickers, sub-types that display different keypads based on field definition, and so. Older devices will still render the

application they will offer less advanced features. The major advantage of developing browser-based applications is that you only need to develop them once for desktop and mobile devices. However, one major limitation is accessing on-device features such as contact lists. This limitation can be alleviated by integrating with solutions such as PhoneGap that support the creation of hybrid solutions, which use a native application wrapper to display the web applications. Solutions such as Phonegap provide various APIs to access many of the native phone features not currently available using HTML5 or JavaScript. To learn more, see:

<http://phonegap.com/>

About Mobile Support

Advantages of the Universal Theme

Universal Theme - 42 (Universal Theme) enables developers to build modern web applications without requiring extensive knowledge of HTML, CSS, or JavaScript

Key advantages of the Universal Theme include:

- **Responsive Design** - Designed to work just as well on small screen devices (such as smartphones and tablets) as it does on larger screen devices (including laptops and desktops). The UI components in Universal Theme work across varying screen resolutions while maintaining the same or similar functionality. In addition, Universal Theme takes full advantage of ultra high screen resolutions by utilizing vector graphics where possible, and relying upon CSS3 features for UI styling.
- **Versatile User Interface** - Provides all the components and building blocks necessary to build practically any type of business application user interface. To browse all of the components provided with Universal Theme go to the *Universal Theme* application at <https://apex.oracle.com/ut> and select **Components**.
- **Easy Customization** - Effortlessly customize and fully control the look and feel of your applications without becoming an expert in UI design, HTML, CSS, or JavaScript. Using Theme Roller and Template Options, you can easily customize your application to fit your company's brand and customize the look and feel of various components using Template Options.

About Migrating Existing Mobile Apps to the Universal Theme

jQuery Mobile and the jQuery Mobile User Interface used in previous releases have been desupported and will be removed in a future release. If you have an existing mobile application that uses the jQuery Mobile User Interface, you should migrate your existing application to the Universal Theme.

Tip:

To learn more about migrating existing applications to the Universal Theme, go to the *Universal Theme* application at <https://apex.oracle.com/ut> and select **Migration Guide**.

Reports Optimized for Mobile Environments

Although the Universal Theme is optimized to work well on mobile devices, not all components are mobile friendly. For example, interactive reports and interactive grids do not work well in mobile environments. When creating reports for mobile devices, Oracle recommends the following report types:

- **List View**

Features a responsive design to display data and provide easy navigation on Smartphones. Creates a page that contains the formatted result of a SQL query. You choose a table on which to build the List view and select a database column to be used for the List view entry.

- **Column Toggle Report**

Creates a responsive report designed for mobile applications and Smartphones. By default, column toggle reports are created with all columns set to the same priority. However, the developer can edit the report column attributes and rank columns by importance. Columns with a lesser priority (larger number) are hidden at narrower screen widths. The report includes a Columns button which enables end users to select which columns they want to view.

- **Reflow Report**

Creates a responsive report designed for mobile applications and Smartphones. When there is not enough space available to display the report horizontally, the report responds by collapsing the table columns into a vertical value pairs layout where each column displays on a separate row.

See Also:

- ["About Switching the Active Theme"](#)
- ["Understanding Template Options"](#)
- ["Using Theme Styles and Theme Roller"](#)

Creating Mobile Applications

To create a mobile application:

1. Create a new application by running the Create Application Wizard. See "[Creating a Database Application Based on a Table or Query](#)."
2. Run the Create Page Wizard and add mobile pages. See "[Adding a New Page to an Application](#)."

Tip:

To learn more about the Universal Theme and mobile-friendly design, go to the *Universal Theme* application at <https://apex.oracle.com/ut>.

Deleting an Application

You can delete an application from the Application home page, or while editing application attributes.

When you delete an application, you are prompted whether to remove the application definition and/or drop supporting objects.

- [Deleting an Application from the Application Home Page](#)
- [Deleting an Application from the Edit Application Definition Page](#)



See Also:

["Deinstalling Supporting Objects"](#) and ["Removing a Productivity and Sample App"](#)

Deleting an Application from the Application Home Page

To delete an application from App Builder:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. On the Application home page, verify the application ID and name at the top of the page.
4. On the Tasks list, click **Delete this Application**.
5. When prompted, click **Permanently Delete Now**.

Deleting an Application from the Edit Application Definition Page

To delete an application from the Edit Application Definition page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Edit Application Properties** button to the right of the application name.
The Edit Application Definition page appears.
4. Verify the application ID and name.
5. Click **Delete** at the top of the page.
6. When prompted, click **Permanently Delete Now**.



See Also:

["Editing Application Attributes"](#)

Managing Application Attributes

Application attributes (also known as the Application Definition) control the behavior of an entire application and are divided into the categories: Definition, Security, Globalization, and User Interface.

Once you create an application, you can review and update all application attributes on the Edit Application Definition page.



Tip:

["How to Create a Custom Application"](#) for information on using the Supporting Objects utility to create a custom application.

- [Editing Application Attributes](#)
Use the Edit Application Definition page to change general application attributes such as the application name, version number, and control various properties and behaviors (such as logging, debugging, feedback, compatibility mode, application availability, error handling, and substitution strings).
- [Configuring Security Attributes](#)
Configure security for all pages in an application on the Edit Security Attributes page. The security attributes are divided into the categories: Authentication, Authorization, Session Management, Session State Protection, Browser Security, and Database Session.
- [Configuring Globalization Attributes](#)
Edit attributes on the Edit Globalization Attributes to configure application globalization options. In App Builder you can develop applications that can run concurrently in different languages.
- [Managing the Application User Interface](#)
Edit attributes on the User Interface page to determines the default characteristics of the application and optimizes the display for the target environment (such as Desktop or Mobile). You can customize the user interface by editing general attributes on the User Interface page and specific attributes on the User Interface Details page.

Editing Application Attributes

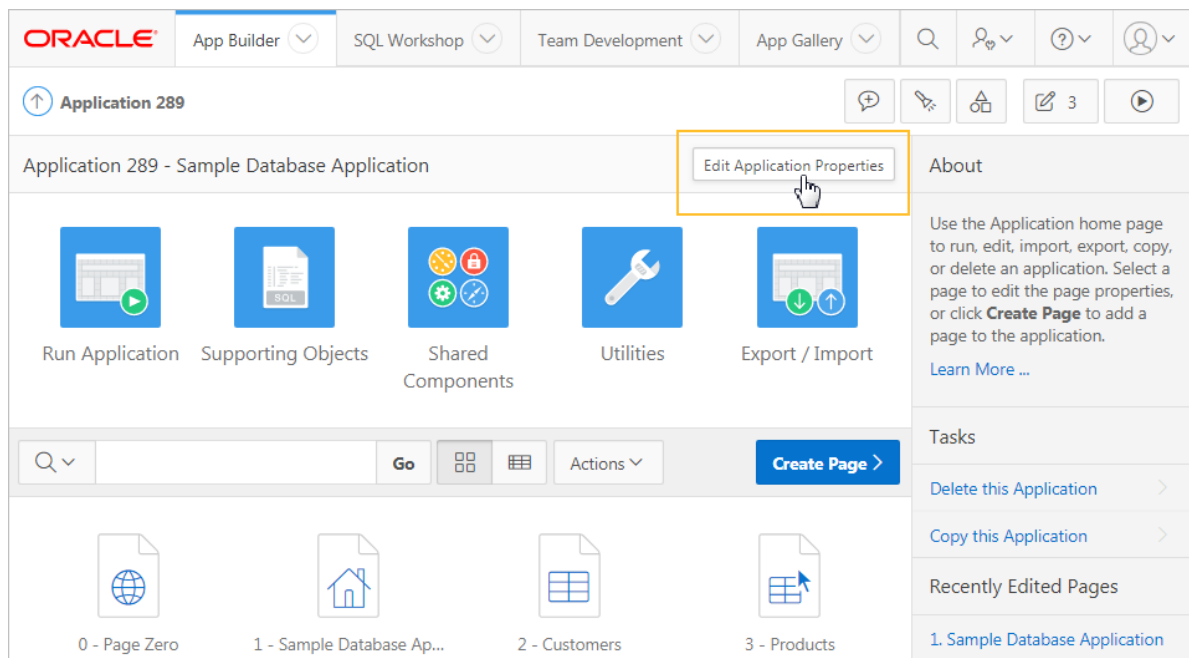
Use the Edit Application Definition page to change general application attributes such as the application name, version number, and control various properties and behaviors (such as logging, debugging, feedback, compatibility mode, application availability, error handling, and substitution strings).

- [Accessing the Edit Application Definition Page](#)
- [Edit Application Definition Page](#)

Accessing the Edit Application Definition Page

To access the Edit Application Definition page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. From Application home page, you can access the Edit Application Definition page by clicking the **Edit Application Properties** button or **Shared Components**.



- From Edit Application Properties:
 - a. Click the **Edit Application Properties** button to the right of the application name.
 - b. Click the **Definition** tab.
- From Shared Components:
 - a. Click **Shared Components**.
 - b. Under Application Logic, click **Application Definition Attributes**.

The Edit Application Definition page appears.

4. Edit the appropriate attributes.
5. Click **Apply Changes** to save your changes.

Edit Application Definition Page

The Edit Application Definition page is divided into the following regions.

**Note:**

Required values are marked with a red asterisk (*).

- [Name](#)
- [Properties](#)
- [Application Icon](#)
- [Availability](#)
- [Error Handling](#)
- [Global Notification](#)
- [Substitutions](#)
- [Build Options](#)

Name

Use Name attributes to define basic characteristics of your application, including the application name, an optional alphanumeric alias, and a version number.

Table 6-4 Application Definition, Name

| Attribute | Description | To Learn More |
|-------------------|--|--|
| Name | A short descriptive name for the application to distinguish it from other applications in your development environment. | n/a |
| Application Alias | Assigns an alternate alphanumeric application identifier. You can use this identifier for the application ID. For example, suppose you create an alias of <code>myapp</code> for application 105. Using <code>f?p</code> syntax, you could call application 105 as either: <ul style="list-style-type: none"> • <code>f?p=105:1</code> • <code>f?p=myapp:1</code> | See " About Using f?p Syntax to Link Pages " |
| Version | Enter the application's version number. The value enter displays on the page. You can also automatically tie the version to the date of last modification using the following format masks: <ul style="list-style-type: none"> • <code>YYYY.MM.DD</code> • <code>MM.DD.YYYY</code> • <code>DD.MM.YYYY</code> If your application version uses <code>YYYY.MM.DD</code> , then App Builder replaces this format mask with the date of last modification of any application attribute. | n/a |

Table 6-4 (Cont.) Application Definition, Name

| Attribute | Description | To Learn More |
|-------------------|---|---|
| Application Group | Displays the application group currently associated with this application. To select another application group, make a selection from the list. To remove an application from an existing group, select Unassigned . | See " Managing Application Groups " |

Properties

Use Properties attributes to configure logging, debug behavior, feedback, compatibility, email from address, and the application proxy server.

Table 6-5 Application Definition, Properties

| Attribute | Description | To Learn More |
|-----------|---|--|
| Logging | <p>Determines whether user activity is recorded in the Oracle Application Express activity log. Select Yes to log every page view and enable an administrator to monitor user activity for each application.</p> <p>Disabling logging may be advisable for high volume applications.</p> <p>This attribute can only be modified if the Application Activity Logging attribute in Oracle Application Express Administration Services is set to Use Application Setting.</p> | See "Enabling Application Activity" in Logging in <i>Oracle Application Express Administration Guide</i> . |
| Debugging | <p>Controls debug mode for the current application. Available options include:</p> <ul style="list-style-type: none"> Yes - Enables the application to run in a debug mode from a browser at runtime. No - Disables the application from enabling in debug from a browser. <p>Running an application in debug mode is useful when an application is under development. For production applications, Oracle recommends disabling debugging and thus preventing users from viewing application logic.</p> <p>Debug can be enabled programmatically regardless of this debug setting. If the application is run from the Application Express development environment, debugging can always be enabled.</p> | n/a |

Table 6-5 (Cont.) Application Definition, Properties

| Attribute | Description | To Learn More |
|--------------------------------|--|---|
| Allow Feedback | <p>Enables support for end user feedback for this application. Select Yes or No.</p> <p>If you select Yes and enable this option, you must create a feedback page and navigation bar icon to call that page. If you later disable feedback, the navigation bar icon is hidden. This enables you to turn on feedback for testing and turn it off for production.</p> | See " Managing Feedback " |
| Compatibility Mode | <p>Controls the compatibility mode of the Application Express runtime engine. Certain runtime behaviors are changed from release to release. Use this attribute to obtain specific application behavior. To realize new behavior in an application, set the compatibility mode of the application to the current version.</p> | n/a |
| Application Email from Address | <p>Determines the email address to use as the from address in the application.</p> <p>Enter a valid email address to use as the from address when sending email from an email download or subscription. The value can be a literal string containing a valid email or a static substitution reference defined in the application using substitution syntax APP_EMAIL. Examples:</p> <pre>john.doe@abc.com &MY_APP_EMAIL_FROM.</pre> <p>Oracle does not recommend using an item substitution at the application or page-level since it only works in email download, but not for subscriptions.</p> <p>Tip: You can also specify the Email from Address by editing interactive report attributes. See "About Emailing from an Interactive Report."</p> | n/a |

Table 6-5 (Cont.) Application Definition, Properties

| Attribute | Description | To Learn More |
|--------------|---|---------------|
| Proxy Server | <p>Specify a proxy server.</p> <p>For example, App Builder may require a proxy server when using a region source type of URL. The URL region source embeds the results of the URL (that is, the page returned by navigating to the URL) as the region source. If you use a firewall and the target of a URL is outside the firewall relative to App Builder, you may need to specify a proxy server.</p> <p>You can reference values entered into this field from PL/SQL using the PL/SQL package variable <code>APEX_APPLICATION.G_PROXY_SERVER</code>.</p> <p>For example: <code>www-proxy.company.com</code></p> | n/a |

Application Icon

In **Icon File Name**, enter the file name of an image which displays as the Application Icon in App Builder. The image should be 64 x 64 pixels, must be uploaded as **Application Image** within Shared Components, and must be present at the 'root' level (that is, not within a folder).



See Also:

["Managing Static Application Files"](#) and ["Managing Static Workspace Files"](#)

Availability

Use Availability attributes to manage your application by defining an application status and build status. For example, if you select the status **Restricted Access**, you can specify which users have access and can run the application.

Table 6-6 Application Definition, Availability

| Attribute | Description | To Learn More |
|-------------------------------------|---|--|
| Status | <p>Specifies whether the application is available or unavailable for use. Options include:</p> <ul style="list-style-type: none"> • Available - Application is available with no restrictions. • Available with Developer Toolbar - Application is available for use. For developers, the Developer Toolbar displays on each page. Requires the developer to be logged in to App Builder in the same browser session. • Available to Developers Only - Application is available to users having developer privileges. • Restricted Access - Application is available to developers named in the Restrict to comma separated user list. • Unavailable - Application cannot be run or edited. The message in Message for unavailable application displays when users attempt to access the application. • Unavailable (Status Shown with PL/SQL) - Application cannot be run or edited. • Unavailable (Redirect to URL) - Application cannot be run. The user is linked to the URL entered in Message for unavailable application. | <p>See:</p> <ul style="list-style-type: none"> • "Changing Build Status for Multiple Applications" and "Changing Application Build Status in Administration Services" in <i>Oracle Application Express Administration Guide</i> • "Controlling Access to Applications, Pages, and Page Components" |
| Build Status | <p>Identifies the build status of the current application. Options include:</p> <ul style="list-style-type: none"> • Run and Build Application - Developers and users can both run and develop the application. • Run Application Only - Users can only run the application. This option is intended for applications in a production instance. | <p>See "Changing Application Build Status Set During Deployment" in <i>Oracle Application Express Administration Guide</i></p> |
| Message for unavailable application | <p>Use this attribute with Status. If you set Status to Unavailable, Unavailable (Status Shown with PL/SQL), or Unavailable (Redirect to URL), the text you enter in this attribute displays. If you set Status to Available, the text you enter in this attribute does not display.</p> | n/a |

Table 6-6 (Cont.) Application Definition, Availability

| Attribute | Description | To Learn More |
|---|--|---------------|
| Restrict to comma separated user list (status must equal Restricted Access) | Use this attribute with the Status Restricted Access . If you set Status to Restricted Access , only the users listed in this attribute can run the application. To use this attribute: <ol style="list-style-type: none"> From the Status list, select a restricted status. Enter a comma-delimited list of users who can run the application in the field provided. Click Apply Changes. | n/a |

Error Handling

Use the Error Handling attributes described to control or modify how an application logs errors.

Tip:

Error handling functions specified here are overridden by similar page-level attributes.

Table 6-7 Application Definition, Error Handling

| Attribute | Description | To Learn More |
|--------------------------------|--|---------------|
| Default Error Display Location | Identifies where the validation error messages display for basic validations performed by Application Express or by plug-ins. Validation error messages can display in a notification area (defined as part of the page template), or within the field label. Options include: <ul style="list-style-type: none"> Inline with Field and in Notification - Error messages display in a notification area defined as part of the page template. Inline with Field - Error messages display within the field label. Inline in Notification - Displays in the <code>#NOTIFICATION_MESSAGE#</code> template substitution string when an error occurs on the page. | n/a |

Table 6-7 (Cont.) Application Definition, Error Handling

| Attribute | Description | To Learn More |
|-------------------------|--|--|
| Error Handling Function | <p>Enter the name of a PL/SQL error function to be called to modify the existing error message and display a more user-friendly message or log the error if one occurs. This function can reference a package function or standalone function in the database. For example:</p> <pre>log_apex_error</pre> <p>When referencing a database PL/SQL package or standalone function, use the #OWNER# substitution string to reference the parsing schema of the current application. For example:</p> <pre>#OWNER#.log_apex_error</pre> <p>You must implement error handling functions using the syntax described in the <code>apex_error</code> package.</p> <pre>function <name of function> (p_error in apex_error.t_error) return apex_error.t_error_result</pre> <p>Note: Error handling specified at the page-level overwrites any error handling function specified here.</p> | See <code>apex_error</code> in <i>Oracle Application Express API Reference</i> |

Global Notification

Use the **Global Notification** attribute to communicate system status to application users. If the page templates used in your application contain the #GLOBAL_NOTIFICATION# substitution string, the text entered here displays in that string's place. For example, you can use this attribute to notify users of scheduled downtime, or communicate other messages regarding application availability.

To create a global notification:

1. Include the #GLOBAL_NOTIFICATION# substitution string in your page template.
2. Navigate to the Edit Application Definition page and enter a message in the Global Notification attribute.
3. Click **Apply Changes**.



See Also:

["Using Substitution Strings"](#) and ["Page Templates"](#)

Substitutions

Use **Substitutions** to define static substitution strings for your application. You can use static substitution string for phrases or labels that occur in many places within an application. To create a substitution string, enter the string name in the **Substitution String** column and the string value in the **Substitution Value** column.

Defining static substitution strings centrally enables you to change text strings in multiple places in your application by making a single change to the **Substitution Value** defined on this page.



See Also:

["Using Substitution Strings"](#)

Build Options

The Build Options displays existing build options in the current application. Most applications have a build option attribute. Build Options have two possible values: `INCLUDE` and `EXCLUDE`. If you specify an attribute to be included, then the Application Express engine includes and enables it at runtime. However, if you specify an attribute to be excluded, then the Application Express engine disables it and excludes it at runtime.

Do not specify a build option unless you plan to exclude that object from specific installations.



See Also:

["Using Build Options to Control Configuration"](#)

Configuring Security Attributes

Configure security for all pages in an application on the Edit Security Attributes page. The security attributes are divided into the categories: Authentication, Authorization, Session Management, Session State Protection, Browser Security, and Database Session.

- [Accessing the Edit Security Attributes Page](#)
- [Security Attributes Page](#)

 **See Also:**
"Managing Application Security"

Accessing the Edit Security Attributes Page

To access the Edit Security Attributes page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. From the Application home page, you can access the Edit Security Attributes page in two ways:
 - From Shared Components:
 - a. Click **Shared Components**.
 - b. Under Security, click **Security Attributes**.
 - From Edit Application Properties:
 - a. Click the **Edit Application Properties** button to the right of the application name.
 - b. Click the **Security** tab.The Edit Security Attributes page appears.
4. Edit the appropriate attributes.
5. Click **Apply Changes** to save your changes.

Security Attributes Page

Use the Edit Security Attributes page to set application-wide security settings. Edit application components directly to manage more granular settings. The Edit Security Attributes page is divided into the following sections:

 **Note:**
Required values are marked with a red asterisk (*).

- [Authentication](#)
- [Authorization](#)
- [Session Management](#)
- [Session State Protection](#)
- [Browser Security](#)
- [Database Session](#)

Authentication

Authentication is the process of establishing users' identities before they can access an application. Although you can define multiple authentication schemes for your application, only one scheme can be current at a time.

Table 6-8 Authentication Attributes

| Attribute | Descriptions | To Learn More |
|-----------------------|---|---|
| Public User | <p>Identifies the Oracle schema (or user) used to connect to the database through the Database Access Descriptor (DAD). Once a user has been identified, the Application Express engine keeps track of each user by setting the value of the built-in substitution string <code>APP_USER</code>.</p> <p>When <code>APP_USER</code> equals this value, the Application Express engine considers the current session to be a "public user" session. The Application Express engine supports the following built-in display conditions:</p> <ul style="list-style-type: none"> • <code>USER_IS_PUBLIC_USER</code> • <code>USER_IS_NOT_PUBLIC_USER</code> <p>If the current application user (<code>APP_USER</code>) equals the value of this attribute, then the user is logged on as a public user. Some applications have public (not logged in) and private (logged in) modes. By determining if the user is the public user, you can conditionally display or hide information.</p> <p>For example, you can show a login button if the user is the public user and a logout link if the user is not a public user. Reference this value using <code>APEX_APPLICATION.G_PUBLIC_USER</code>.</p> | <p>See "HOME_LINK" and "Understanding Conditional Rendering and Processing"</p> |
| Authentication Scheme | <p>identifies the current authentication method used by this application. The purpose of authentication is to determine the application users identity. To create an authentication scheme, click Define Authentication Schemes.</p> | <p>See "How Authentication Works" and "Creating an Authentication Scheme"</p> |

Authorization

Application authorization schemes control access to all pages within an application. Unauthorized access to the application, regardless of which page is requested, causes an error page to display.

Table 6-9 Authorization Attributes

| Attribute | Descriptions |
|------------------------------|---|
| Authorization Scheme | Specify an authorization scheme for your application. Application authorization schemes are defined for an application for the purpose of controlling access. Setting a required authorization scheme here at the application level will require all pages of the application to pass the defined authorization check. |
| Define Authorization Schemes | Click the Define Authorization Schemes button to create an authorization scheme. |
| Run on Public Pages | Controls whether the application-level authorization scheme is checked on public pages (that is, pages that do not require authorization). Options include: <ul style="list-style-type: none">• Yes - If you select Yes and the page is public, the application authorization is checked.• No - If you select No and the page is public, the application authorization is ignored. Note: This attribute is ignored if you select Must Not Be Public User as Authorization Scheme. |



See Also:

["Attaching an Authorization Scheme to an Application"](#)

Session Management

Use Session Management attributes to reduce exposure by application to abandoned computers with an open web browser.

Table 6-10 Session Management

| Attribute | Descriptions |
|-----------------|---|
| Rejoin Sessions | <p>Use this attribute to control if Application Express should support application URLs that do not contain session IDs. When Rejoin Sessions is enabled, Application Express attempts to use the session cookie to join an existing session, when a URL does not contain a session ID.</p> <p>A more restrictive instance-level setting overrides this page-level value.</p> <p>Note: Enabling rejoin sessions exposes your application to possible security breaches, as it can enable attackers to take over existing end user sessions. See "About Rejoin Sessions."</p> <p>Rejoin Sessions options include:</p> <ul style="list-style-type: none"> • Application Default - The application-level setting applies. • Disabled - If the URL does not contain a session ID Application Express creates a new session. • Enabled for Public Sessions - If the URL goes to a public page and does not contain a session ID, Application Express attempts to utilize the existing session cookie established for that application. Application Express only joins using the cookie when the session is not yet authenticated. • Enabled for All Sessions - If the URL does not contain a session ID, Application Express attempts to utilize the existing session cookie established for that application, providing the following conditions are met: Session State Protection is enabled for the application and the URL includes a valid checksum. For public bookmarks, the most restrictive item level protection must be either Unrestricted or Checksum Required - Application Level. OR, the URL does not contain payload (a request parameter, clear cache or data value pairs). This option requires you set Embed In Frames to Allow from same origin or Deny. This is not tied to a condition about the URL payload, but also applies to session state protected URLs. |
| Deep Linking | <p>Enable or prevents deep linking to an application. Options include:</p> <ul style="list-style-type: none"> • Enabled - The URL to a specific page ultimately redirects there, possibly after the user has logged in. • Disabled - If the URL does not contain a valid session ID, Application Express starts a new session and redirects to the application's home page. <p>For example, browsers often save the URLs of opened tabs and try to restore the sessions after a restart, causing a deep link. This behavior may be undesirable (for example if a URL points to a page in the middle of a multi-step wizard). By selecting Disable, Application Express starts a new session and redirects to the application's home page.</p> |

Table 6-10 (Cont.) Session Management

| Attribute | Descriptions |
|--------------------------------------|--|
| Maximum Session Length in Seconds | <p>Defines how long (in seconds) sessions can exist and be used by this application.</p> <ul style="list-style-type: none"> • Enter a positive integer to control how long a session can exist. • Enter 0 to have sessions exist indefinitely. • Leave the value empty in order to use the session duration defined at the workspace-level or the instance level. |
| Session Timeout URL | <p>Enter an optional URL to redirect to when the maximum session lifetime has been exceeded. The target page in this URL, if implemented in Application Express, should be a public page.</p> <p>A common use for this page would be to inform the user of the session expiration and to present a login link or other options. If you do not enter a URL, users will see the message "Your session has timed out" and a link to the application home page. If you enter #LOGOUT_URL#, Application Express will execute a logout, just like when the user clicked on the application's logout link.</p> <p>Only three substitution items are supported:</p> <ul style="list-style-type: none"> • &APP_SESSION. • &SESSION. • &APP_ID. <p>Because of the particular purpose of this URL, it is not necessary to include either &APP_SESSION. or &SESSION. in the link.</p> |
| Maximum Session Idle Time in Seconds | <p>The Session Idle Time is the time between the last page request and the next page request. Options include:</p> <ul style="list-style-type: none"> • Enter a positive integer to control the idle time for sessions used by this application. • Leave the value empty in order to use the idle time defined at the workspace level or the instance level. |
| Session Idle Timeout URL | <p>Enter an optional URL to be redirected to when the maximum session idle time has been exceeded. The target page in this URL, if implemented in Application Express, should be a public page. A common use for this page would be to inform the user of the session expiration and to present a login link or other options. If you do not enter a URL, users will see the message "Your session has timed out" and a link to the application home page. If you enter #LOGOUT_URL#, Application Express will execute a logout, just like when the user clicked on the application's logout link..</p> <p>Only three substitution items are supported in this URL:</p> <ul style="list-style-type: none"> • &APP_SESSION. • &SESSION. • &APP_ID. <p>Because of the particular purpose of this URL, it is not necessary to include either &APP_SESSION. or &SESSION. in the link.</p> |

 **See Also:**

"[About Utilizing Session Timeout](#)" and "Configuring Session Timeout" in *Oracle Application Express Administration Guide*

Session State Protection

Enabling Session State Protection can prevent hackers from tampering with URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy. This table describes the attributes available under Session State Protection.

Table 6-11 Session State Protection

| Attribute | Descriptions |
|---------------------------------|--|
| Session State Protection | Make a selection from the Session State Protection list, to enable or disable Session State Protection for your application. Selecting Enabled turns on session state protection controls defined at the page and item-level. To learn more, see field-level Help. |
| Allows URLs Created After | Lists the date and time after which bookmarked links are usable to access pages in this application if the bookmarked link contains a checksum and Session State Protection is enabled for the application. Bookmarks created before this date and time are not usable to access this application if the bookmarked link contains a checksum and Session State Protection is enabled for the application. Bookmarks that do not contain checksums or bookmarks that contain checksums that are unnecessary are not affected by this attribute. Their usability is determined using other criteria. A hidden application attribute (a checksum salt) is used during the computation and later verification of checksums included in f?p= URLs generated during page rendering. Checksums are included when Session State Protection is enabled for the application. You can reset this checksum salt attribute at any time by clicking the Expire Bookmarks button. |
| Bookmark Hash Function | Used to create checksums for application-level and user-level checksums in bookmarkable URLs. |
| Expire Bookmarks | Click Expire Bookmarks to reset this hidden application attribute (a checksum salt) salt attribute at any time. Clicking this button causes any bookmarked URLs that contain previously generated checksums to fail when they are subsequently used to access the application. Tip: You can also click Expire Bookmarks to change the Bookmark Hash Function to switch to a different algorithm for computing checksums. |
| Manage Session State Protection | Click Manage Session State Protection to configure Session State Protection. |

**See Also:**["Preventing URL Tampering"](#)

Browser Security

This table describes the attributes available under Browser Security.

Table 6-12 Browser Security

| Attribute | Descriptions |
|--------------------|---|
| Cache | <p>Use Cache to enable or disable browser caching of application page contents. If enabled, the browser saves the contents of pages for this application in its cache, both in memory and on disk. Typically when caching is enabled and the browser back button is clicked, the page is loaded from the cache instead of from the server. If disabled, the browser is instructed not to save application page contents and requests the latest page content from the server whenever the URL changes.</p> <p>To avoid the possibility of saving sensitive data, Oracle recommends that this attribute be disabled. Otherwise, it is possible to go back in the browser history after a logout and see cached content from a previous session. Disabling the browser cache also prevents issues with pages that use partial page refreshes, such as is the case with interactive reports.</p> <p>If this attribute is set to Disabled, Application Express sends the HTTP header <code>cache-control: no-store</code> which instructs the browser to not cache the page contents on disk or in memory. Note that this feature requires modern browsers that support the HTTP header response variable <code>cache-control</code>.</p> |
| Embed in Frames | <p>Controls if a browser may display your application's pages within a frame. Available options include:</p> <ul style="list-style-type: none"> • Deny - The page cannot be displayed in a frame, regardless of the site attempting to do so. • Allow from same origin - The page can only be displayed in a frame on the same origin as the page itself. • Allow - The page can be displayed in any frame. <p>Displaying pages within frames can be misused with "clickjacking" attacks. In a "clickjacking" attack, the attacker uses multiple layers to trick a user into clicking a button or link on another page when they were intending to click the top level page. Thus, the attacker is hijacking clicks (or keystrokes) meant for their page and routing them to another page.</p> <p>To learn more, see field-level Help.</p> |
| HTML Escaping Mode | <p>Defines how Oracle Application Express escapes special characters. Options include:</p> <ul style="list-style-type: none"> • Basic: Escape <code>&</code>, <code>"</code>, <code><</code> and <code>></code> • Extended: Escape <code>&</code>, <code>"</code>, <code><</code>, <code>></code>, <code>'</code>, <code>/</code> and non-ASCII characters if the database character set is not AL32UTF8 |

Table 6-12 (Cont.) Browser Security

| Attribute | Descriptions |
|-----------------------|---|
| HTTP Response Headers | <p>Enter additional application specific HTTP headers that Oracle Application Express should send on each response and that it does not support in another way (for example, X-Frame-Options using the Embed in Frames attribute).</p> <p>Example:</p> <pre>X-XSS-Protection: 1; mode=block X-Content-Type-Options: nosniff</pre> |

 **Tip:**

Both **Cache** and **Embed in Frames** require modern browsers that support the HTTP header response variable X-Frame-Options.

Database Session

This table describes the attributes available under Database Session.

Table 6-13 Database Session

| Attribute | Descriptions |
|----------------------------|---|
| Parsing Schema | Specifies the schema that all SQL and PL/SQL in the application will be parsed as. You may use #OWNER# to reference this value in SQL queries and PL/SQL (for example, in a region or a process). |
| Initialization PL/SQL Code | Use this attribute to enter a PL/SQL block that sets a context for the database session associated with the current "show page" or "accept page" request. The block you enter here is executed at a very early point during the page request, immediately after the APP_USER value is established. The value of APP_USER (using :APP_USER or v('APP_USER')) may be used within the block. Values of other items in session state may be referenced as well, but any such items must have been established in session state before the initiation of the current page request. To view examples, see field-level Help. |
| Cleanup PL/SQL Code | Use this attribute to enter a PL/SQL block that runs at the end of page processing. It can be used to free or clean up resources that were used, like VPD contexts or database links. To view examples, see field-level Help. |

Table 6-13 (Cont.) Database Session

| Attribute | Descriptions |
|-------------------|--|
| Runtime API Usage | Control how this application can access Oracle Application Express APIs that modify applications and workspace data, while it is running. Options include: <ul style="list-style-type: none">• Modify This Application: The application can modify itself.• Modify Other Applications: The application can change other applications in the workspace.• Modify Workspace Repository: The application can change workspace users and groups. |

**See Also:**

"[Providing Security Through Authorization](#)" and *Oracle Label Security Administrator's Guide*

Configuring Globalization Attributes

Edit attributes on the Edit Globalization Attributes to configure application globalization options. In App Builder you can develop applications that can run concurrently in different languages.

A single application can be translated to support different languages. Use the attributes on the Edit Globalization Attributes page to specify globalization options such as the primary application language and defaults for date format, time format, timestamp format, time zone format, and CSV encoding.

- [Accessing the Globalization Attributes Page](#)
- [Edit Globalization Attributes Page](#)

**See Also:**

"[Managing Application Globalization](#)"

Accessing the Globalization Attributes Page

To access the Edit Globalization Attributes page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. Click **Shared Components**.

The Shared Components page appears.

4. Under Globalization, click **Globalization Attributes**.

The Edit Globalization Attributes page appears.

5. Edit the appropriate attributes and click **Apply Changes**.

 **Tip:**

You can also access to the Edit Globalization Attributes page by navigating to the Edit Application Definition and then clicking the Globalization tab. See "[Accessing the Edit Application Definition Page](#)."

Edit Globalization Attributes Page

The following sections describe the attributes available on the Edit Globalization Attributes page.

 **Note:**

Required values are marked with a red asterisk (*).

- [Application Primary Language](#)
- [Application Language Derived From](#)
- [Document Direction](#)
- [Application Date Format](#)
- [Application Date Time Format](#)
- [Application Timestamp Format](#)
- [Application Timestamp Time Zone Format](#)
- [Character Value Comparison](#)
- [Character Value Comparison Behavior](#)
- [Automatic Time Zone](#)
- [Automatic CSV Encoding](#)

 **See Also:**

["Specifying the Primary Language for an Application"](#)

Application Primary Language

Identifies the language in which an application is developed. This language is the base language from which all translations are made. For example, suppose application 100

was authored in English, translated into French, and published as application 101. English would be the Application Primary Language.

All modifications to the application should be made to the primary language specified here.

Application Language Derived From

Specifies how Application Express derives the translated application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item.

To learn more about the available options, see Field-level Help.

Document Direction

Sets the document direction. Options include:

- Left-To-Right
- Right-To-Left

Application Date Format

Determines the date format to be used in the application.

Use this date format to alter the `NLS_DATE_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If no value is specified, the default date format is derived from the database session at runtime. Consider the following examples:

```
Month DD, YYYY  
&MY_DATE_FORMAT.
```

Application Date Time Format

Specify the date time format to be used in the application.

This date time format can be referenced in an application using the substitution reference `&APP_DATE_TIME_FORMAT.`, or in PL/SQL using the function `v('APP_DATE_TIME_FORMAT')`. This attribute does not alter any NLS settings. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If this attribute value is not specified, then a reference to `APP_DATE_TIME_FORMAT` returns the NLS database session date format and the NLS time format. Consider the following examples:

```
Month DD, RRRR HH24:MI  
&MY_DATE_TIME_FORMAT.
```

Application Timestamp Format

Determines the timestamp format to be used in the application. Select a timestamp format from the list of values.

Use this timestamp format to alter the `NLS_TIMESTAMP_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle timestamp format mask or an item reference

using substitution syntax. If no value is specified, the default timestamp format is derived from the database session at runtime. Consider the following examples:

```
DD-MON-RR HH.MI.SSXF AM  
&MY_TIMESTAMP_FORMAT.
```

Application Timestamp Time Zone Format

Determines the timestamp with time zone format to be used in the application.

Use this date format to alter the `NLS_TIMESTAMP_TZ_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle timestamp with time zone format mask or an item reference using substitution syntax. If no value is specified, the default timestamp with time zone format is derived from the database session at runtime. Consider the following examples:

```
DD-MON-RR HH.MI.SSXF AM TZR  
&MY_TIMESTAMP_TZ_FORMAT.
```

Character Value Comparison

Determines the collating sequence for character value comparison in various SQL operations and clauses, for example, `ORDER BY`, `LIKE`, `MIN/MAX`.

Use this value to alter `NLS_SORT` database session parameter for the execution of SQL queries in classic report and interactive report regions. If no value is specified, the default value is derived from the database session at runtime. Consider the following examples:

```
BINARY  
GERMAN  
CANADIAN_M
```

Character Value Comparison Behavior

Determines the collation behavior of SQL operations, for example, `LIKE`, `MIN/MAX`.

This value is used to alter `NLS_COMP` database session parameter for the execution of SQL queries in classic report, interactive report, and List view regions, as well as in plug-in API. Options include:

- **Database session NLS setting (default)** - The `NLS_COMP` value is derived from the database session at runtime.
- **Binary** - Comparisons in `WHERE` clauses and other SQL operations are binary.
- **Linguistic** - Comparisons in `WHERE` clauses and other SQL operations use the linguistic sort specified in the Character Value Comparison attribute (`NLS_SORT`).

Automatic Time Zone

Controls the setting of the database session time zone. When set to **Yes**, the client time zone is derived from the client's web browser and set for the duration of the Application Express session.

Subsequent page views have the database session time zone set properly per page view. Once set, this setting can be overridden using

APEX_UTIL.SET_SESSION_TIME_ZONE, or reset using
APEX_UTIL.RESET_SESSION_TIME_ZONE.



See Also:

Oracle Application Express API Reference

Automatic CSV Encoding

Automatic CSV Encoding controls the encoding of all comma-delimited (CSV) report output in an application. The default value for Automatic CSV Encoding is **Yes**. When Automatic CSV Encoding is set to **Yes**, CSV report output is converted to a character set compatible with localized desktop applications. The character set for the CSV encoding is determined by the Application Language Derived From setting.

The encoding of pages in App Builder is determined by the character set of the Database Access Descriptor (DAD) used to access Oracle Application Express. For example, if the character set of the Database Access Descriptor is AL32UTF8, all pages in all applications in the Oracle Application Express user interface are encoded in UTF-8.

By default, the CSV output from report regions is encoded in the same character set as the Database Access Descriptor. However, some desktop spreadsheet applications require that the data is encoded in the client desktop operating system character set. In the case of multibyte data, the CSV output from report regions often appears corrupted when opened by a desktop spreadsheet application. This is because the CSV output from report regions is encoded differently than what is required by the desktop application. Enabling Automatic CSV Encoding resolves this issue.

For example, if the user's language preference for an application is `de`, the CSV data is encoded in Western European Windows 1252, regardless of the Database Access Descriptor character set setting. If the user's language preference is `zh-cn`, the CSV data is encoded in Chinese GBK.

Managing the Application User Interface

Edit attributes on the User Interface page to determine the default characteristics of the application and optimize the display for the target environment (such as Desktop or Mobile). You can customize the user interface by editing general attributes on the User Interface page and specific attributes on the User Interface Details page.

- [Accessing User Interface Attributes](#)
- [User Interface Page](#)
- [Defining an Application Logo](#)
- [Adding Desktop UI to a Mobile UI Only Application](#)
- [Editing User Interface Details](#)
- [User Interface Details Page](#)

Accessing User Interface Attributes

To access user interface attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

The Application home page appears.

3. From the Application home page, you can access the User Interfaces page in two ways:
 - From Shared Components:
 - a. Click **Shared Components**.
 - b. Under User Interface, click **User Interface Attributes**.
 - From Edit Application Properties:
 - a. Click the **Edit Application Properties** button to the right of the application name.
 - b. Click the **User Interface** tab.

The User Interfaces page appears.

4. To edit attributes for a specific user interface, select a previously defined user interface in User Interfaces region.

The User Interfaces page appears. Defined User Interfaces display in the **User Interfaces** region at the top of the page.

5. Edit the appropriate attributes.
6. Click **Apply Changes** to save your changes.



See Also:

["Editing User Interface Details"](#)

User Interface Page

The User Interface page is divided into the following sections:



Tip:

To learn more about the attributes on this page, see field-level Help.

- [User Interfaces](#)
- [General Properties](#)
- [Logo](#)
- [Favicon](#)

- [User Interface Detection](#)

User Interfaces

Displays user interfaces defined for the current application. To edit an existing user interface, click the user interface name. The User Interface Details page appears.

For applications with only a Mobile User Interface, developers can add the Desktop User Interface. To add a new Desktop User Interface, click **Add New User Interface** and edit the attributes.

See Also:

- ["Accessing User Interface Attributes"](#)
- ["Adding Desktop UI to a Mobile UI Only Application"](#)
- ["User Interface Details Page"](#)

General Properties

Use General Properties to define basic characteristics of the application user interface.

Table 6-14 User Interface, General Properties

| Attribute | Description |
|--------------------|--|
| Static File Prefix | <p>Determines the virtual path the Web server uses to point to the static files when using the #APP_IMAGES# substitution string.</p> <p>Do not specify anything to reference files which are stored with your application definition in the database.</p> <p>For performance reasons you can also store your application files on your Web Server. Use any valid URL to reference them.</p> <p>Examples:</p> <ul style="list-style-type: none"> • /myFiles/ • http://contentDeliveryNetwork.com/myFiles/ |
| Image Prefix | <p>Determines the virtual path the web server uses to point to the images directory distributed with App Builder. During installation, the virtual path is configured as /i/. If you are unsure, contact your administrator.</p> <p>When embedding an image in static text (for example, in page or region headers or footers), you can reference an image using the substitution string #IMAGE_PREFIX#. For example, to reference the image go.gif, you would use the following syntax:</p> <pre></pre> <p>Note: If the entered image prefix is equal to the instance image prefix, then the application-level attribute will always be null. This easily facilitates the movement of an application across different instances that may have different image prefixes.</p> <p>See Also: "IMAGE_PREFIX"</p> |

Table 6-14 (Cont.) User Interface, General Properties

| Attribute | Description |
|------------|---|
| Media Type | <p>Enter the Internet media type. An Internet media type is two-part identifier for file formats on the internet. A Media Type is composed of at least two parts: a type, a subtype, and one or more optional parameters. This Media Type is used in the Content-Type HTTP header when rendering the page.</p> <p>The page-level Media Type overrides the application-level Media Type. The default value for this attribute is NULL. If both the page-level and application-level values for Media Type are NULL, the Media Type <code>text/html</code> is used.</p> |

Logo

Use Logo attributes to define an application logo. An application logo can be text-based or image-based. To use this feature, your page template must include the `#LOGO#` substitution string.

Table 6-15 User Interface, Logo

| Attribute | Description |
|-----------------|---|
| Logo Type | <p>For Logo Type, select an image type:</p> <ul style="list-style-type: none"> Select Image to use an image for the application logo. Select Text to use text for the application logo. |
| Logo | <p>For Logo Type, select an image type:</p> <ul style="list-style-type: none"> For an image, enter the complete image name, including the filename extension, for example: <code>/i/oracle.gif</code> For text, enter the full text string, for example: <code>Sample Application</code> <p>Note that to use this feature, your page template must include the <code>#LOGO#</code> substitution string.</p> |
| Logo Attributes | <p>In Logo Attributes, enter the attributes for the logo.</p> <p>Image example: <code>width="100" height="20" alt="Company Logo"</code></p> <p>Text example: <code>style="font-family:Arial; color:#000000; font-size:18; white-space:nowrap; font-weight:bold;"</code></p> <p>Note that to use this feature, your page template must include the <code>#LOGO#</code> substitution string.</p> |



See Also:

["Defining an Application Logo"](#)

Favicon

Enter Favicon HTML code in this attribute to create a favicon (or shortcut icon). To use this feature, your page template must include the #FAVICON# substitution string.

Example:

```
<link rel="shortcut icon" href="/i51/favicon.ico">
<link rel="icon" sizes="16x16" href="/i51/favicon-16x16.png">
<link rel="icon" sizes="32x32" href="/i51/favicon-32x32.png">
<link rel="apple-touch-icon" sizes="180x180" href="/i51/
favicon-180x180.png">
```

User Interface Detection

Enter **CSS file URLs** for stylesheets that should be loaded when Application Express displays the available user interfaces, when automatic detection fails. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string #MIN# to include .min , or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string #APP_VERSION# if you want to include the application's version in the file URL.

To view examples, see field-level Help.

Defining an Application Logo

To define an application logo:

1. Access the User Interface page.
2. Specify the logo attributes.
 - If the logo is an image:
 - a. Logo Type - Select **Image**.
 - b. Logo - Enter the complete image name, including the file name extension (for example, /i/oracle.gif) or a fully qualified URL if you are referencing the image.
 - c. Logo Attributes - Enter the appropriate attributes for the logo or make a selection from the list.
 - If the logo is text:
 - a. Logo Type - Select **Text**.
 - b. Logo - Enter the full text string, for example:
Sample Application
 - c. Logo Attributes - Enter the appropriate attributes for the logo or make a selection from the list.
3. Click **Apply Changes**.

 **Tip:**

You can also reference an image uploaded to the static file repository using a substitution string. See "[Referencing Static Application Files](#)" and "[About Referencing Static Workspace Files](#)."

 **See Also:**

- "[Accessing User Interface Attributes](#)"
- "[Managing Static Application Files](#)"
- "[Managing Static Workspace Files](#)"
- "[Creating Custom Themes](#)"
- "[Page Templates](#)"

Adding Desktop UI to a Mobile UI Only Application

For applications with only a Mobile User Interface, developers can add the Desktop User Interface to facilitate the migration from the desupported Mobile User Interface to the Desktop User Interface and the Universal Theme.

 **Note:**

Oracle recommends rebuilding the Mobile UI pages using the Desktop UI and then once functionally complete, removing the pages associated with the Mobile UI and deleting the Mobile UI from the application. To learn more about migrating existing applications to the Universal Theme, go to the *Universal Theme* application at <https://apex.oracle.com/ut>.

To add a new user interface:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. From the Application home page, you can access the User Interfaces page in two ways:
 - From Shared Components:
 - a. Click **Shared Components**.
 - b. Under User Interface, click **User Interface Attributes**.
 - From Edit Application Properties:

- a. Click the **Edit Application Properties** button to the right of the application name.
- b. Click the **User Interface** tab.

The User Interfaces page appears.

4. In the User Interfaces region, click **Add New User Interface**.
5. On User Interface:
 - a. Type - Select a user interface type (for example, **Mobile**). If no types display, then all available user interfaces are associated with the current application.
 - b. Review the remaining attributes.
To learn more about an attribute, see field-level Help.
 - c. Click **Next**.
6. On Create Theme:
 - a. Theme Type - Select a theme type. Standard themes are supplied with Application Express. Custom themes are additional themes available for use.
 - b. Theme - Select a theme (for example, **Mobile (Theme 51)**).
 - c. Click **Next**.
7. Confirm your selections and click **Create**.

The user interface is added to the User Interface page. You can edit the user interface by editing User Interface Details.

Editing User Interface Details

Use User Interface Details to define the specific settings for a selected user interface type. For example, you can select the application home and login page, configure the Navigation Menu and Navigation Bar as well as configure attributes for JavaScript, Cascading Style Sheet, and Concatenated Files.

To edit user interface details:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. From the Application home page, you can access the User Interfaces page in two ways:
 - From Shared Components:
 - a. Click **Shared Components**.
 - b. Under User Interface, click **User Interface Attributes**.
 - From Edit Application Properties:
 - a. Click the **Edit Application Properties** button to the right of the application name.
 - b. Click the **User Interface** tab.

The User Interfaces page appears. Defined User Interfaces display in the **User Interfaces** region at the top of the page.

4. To edit attributes for a specific user interface, click the Edit icon adjacent to the User Interface.
The User Interface Details page appears.
5. Edit the appropriate attributes. To learn more about an attribute, see field-level Help.
6. Click **Apply Changes** to save your changes.

User Interface Details Page

Use User Interface Details to define the specific settings for the selected user interface type.

The User Interface Details page is divided into the following sections:

- [Identification](#)
- [Attributes](#)
- [Navigation Menu](#)
- [Navigation Bar](#)
- [JavaScript](#)
- [Cascading Style Sheets](#)
- [Concatenated Files](#)



See Also:

["Accessing User Interface Attributes"](#)

Identification

Table 6-16 User Interface Details, Identification

| Attribute | Description |
|---------------------|---|
| User Interface Type | Displays the selected interface type. A Desktop user interface is used for applications primarily designed for desktop use. A Mobile user interface is used for applications primarily designed for use on smartphones and tablets. |
| Display Name | Specify a display name for the user interface. The display names is shown in wizards, such as the Create Theme Wizard. |
| Sequence | Specify the display sequence for the user interface. |

Attributes

Table 6-17 User Interface Details, Attributes

| Attribute | Description | To Learn More |
|--|---|--|
| Auto Detect | Select whether the user interface should be automatically detected. If auto-detection is enabled, the user will be redirected to the corresponding login page or home page. | n/a |
| Default | Select whether the user interface is the default interface for the application. | n/a |
| Enable End Users to choose Theme Style | If set to Yes , end users can choose a Theme Style for their sessions within a customization dialog. Only Theme Styles marked as Public are eligible for selection. | See "Enabling Users to Select a Theme Style" |
| Add "Built with APEX" to Footer | If set to Yes , Oracle Application Express will add the text "Built with ♥ using Oracle APEX" to the footer of every page. | n/a |
| Home URL | Specify the home page of the application for the current user interface. | n/a |
| Login URL | Specify the login page of the application for the current user interface. | n/a |
| Theme | Shows the theme currently associated with the user interface. | See "Switching Themes" |
| Theme Style | Select a theme style. This option only displays for newer themes that support theme styles. | See "Using Theme Styles and Theme Roller" |
| Global Page | If defined, displays the global page for the application. | See "Creating a Global Page to Display Components on Every Page" |

Navigation Menu

Applications using newer themes, such as *Universal Theme - 42*, provide navigation with navigation menus.

Table 6-18 User Interface Details, Navigation Menu

| Attribute | Description |
|----------------------|---|
| Display Navigation | Select Yes or No . |
| Navigation Menu List | Select the list utilized for the navigation menu for the application. |

Table 6-18 (Cont.) User Interface Details, Navigation Menu

| Attribute | Description |
|------------------|--|
| Position | <p>Select the position where you would like your navigation menu to be placed on this application.</p> <ul style="list-style-type: none"> • Side renders the navigation menu list template in the #SIDE_GLOBAL_NAVIGATION_LIST# position on your page template. • Top renders the navigation menu list template in the #TOP_GLOBAL_NAVIGATION_LIST# position on your page template. <p>For example, if you choose Side, your navigation could be rendered as a tree on the left side of your pages; and choosing Top can render your navigation as a menu bar in the header of your pages.</p> |
| List Template | Select the List Template used to render the navigation menu for this application. |
| Template Options | Set Template Options for the List Template used for the navigation menu list for the application. |

**See Also:**

["Managing Navigation Menus"](#)

Navigation Bar

Applications using newer themes, such as *Universal Theme - 42*, include navigation bar lists. Navigation Bar settings enable you to select list and list templates. Selecting classic implementation uses tabs instead of a list. Navigation Bar attributes only display with newer themes.

Table 6-19 User Interface Details, Navigation Bar

| Attribute | Description |
|---------------------|---|
| Implementation | <p>Select how to you wish to implement the navigation bar in this application.</p> <ul style="list-style-type: none"> • Classic renders the navigation bar as a classic navigation bar in the #NAVIGATION_BAR# position on your page template. • List renders the navigation bar as a list, using the selected list and list template in the #NAVIGATION_BAR# position on your page template. |
| Navigation Bar List | Select the list utilized for the navigation bar on this application. |
| List Template | Select the List Template used to render the navigation menu for this application. |
| Template Options | Set Template Options for the List Template used for the navigation menu list for the application. |

**See Also:**["Managing Navigation Bar Lists"](#)

JavaScript

Use these attributes to control or modify how an application handles JavaScript.

Table 6-20 User Interface Details, JavaScript

| Attribute | Description |
|--|---|
| Content Delivery Network | <p>Specify the Content Delivery Network (CDN) that Application Express will try to use to load the libraries jQuery and jQuery Mobile. If Application Express cannot load these libraries from the CDN, they will be loaded from your web server instead.</p> <p>Using a CDN can reduce the loading time of your application if the user has already visited other web sites which also use the same CDN to load the same libraries.</p> |
| File URLs | <p>Enter JavaScript file URLs for code to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file, you can use the substitution string #MIN# to include .min or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode.</p> <p>JavaScript file URLs you enter here replaces the #APPLICATION_JAVASCRIPT# substitution string in the page template.</p> <p>Note: You do not need to include opening or closing script tags. Just write the URL.</p> <p>Examples:</p> <ul style="list-style-type: none"> Standard file reference: <pre>/myjs/main.js</pre> Standard file reference which loads the minified file main.min.js for regular page views and main.js in Debug mode: <pre>/myjs/main#MIN#.js</pre> Conditional file for Internet Explorer <pre>[if IE]/myjs/ie.js</pre> |
| Include Deprecated or Desupported Javascript Functions | <p>Specifies if deprecated or desupported JavaScript functions are included on every page in the application. The functions deprecated or desupported are listed in the Release Notes for every release. If you are confident your application does not contain any references to those deprecated or desupported functions, set this to No to reduce the overall size of the JavaScript files loaded.</p> <p>See Also: "Legacy JavaScript APIs" in <i>Oracle Application Express API Reference</i></p> |

Table 6-20 (Cont.) User Interface Details, JavaScript

| Attribute | Description |
|------------------------|--|
| Include jQuery Migrate | <p>Specifies if the jQuery Migrate plug-in should be included on every page in the application.</p> <p>The plug-in restores deprecated features and behaviors of jQuery so that old JavaScript code and jQuery plug-ins will still run properly with the jQuery version loaded by Application Express.</p> <p>If you are confident your application and any used jQuery plug-in does not contain any references to deprecated jQuery features, set this to No to reduce the overall size of the JavaScript files loaded.</p> |

Cascading Style Sheets

In **File URLs**, enter Cascading Style Sheet file URLs to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.

File URLs you enter here will replace the `#APPLICATION_CSS#` substitution string in the page template.

For examples, see field-level Help.

Concatenated Files

Using a concatenated file can increase the performance of loading your page because instead of issuing multiple HTTP requests for each single file, the browser only loads one file. This approach gives you the option to use smaller, more modular files during development and to use a single concatenated file when running the application outside of the Application Express development environment.

To create a concatenated file, click **Concatenated File** and follow the on-screen instructions. To learn more and view examples, see field-level Help.

Adding Developer Comments

Add comments to an application, a page, or a group of pages using the Developer Comment, Bug, or To Do button. You can use developer comments to communicate application changes, report issues, or record developer suggestions.

- [Developer Comment, Bug, or To Do Button](#)
- [Adding Developer Comments to an Application](#)
- [Viewing and Editing Developer Comments](#)
- [Deleting Developer Comments](#)
- [Accessing the Developer Comments from Shared Components](#)

- [Viewing the Developer Comments Calendar](#)

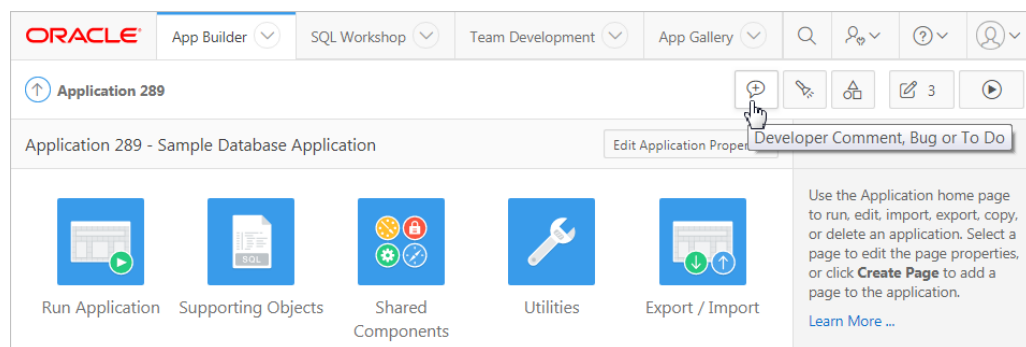


See Also:

["Adding Database Application Comments"](#)

Developer Comment, Bug, or To Do Button

The Developer Comment, Bug, or To Do button resembles a small word balloon. This button displays on most pages in App Builder that relate to a specific application or application page.



Adding Developer Comments to an Application

You can add developer comments to an application, a page, or a group of pages.

To add a developer comment:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment, Bug, or To Do** button.
4. From the Type list, select **Developer Comment**.
5. In Comment, enter up to 4000 characters of text.
6. Click **Create**.

Viewing and Editing Developer Comments

To edit a developer comment:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment, Bug, or To Do** button.
4. Click **View Comments**.

The Developer Comments page appears.

5. To edit a comment, click the **Edit** icon.
The Edit Comment page appears.
 - a. In Comment, enter up to 4000 characters of text.
 - b. Click **Apply Changes**.

 **Tip:**

You can also view and edit Developer comments on the Shared Components page. Navigate to the Shared Components page and click **Developer Comments** in the Tasks region. See "[Accessing the Developer Comments from Shared Components](#)."

Deleting Developer Comments

You can delete specific developer comments or delete multiple comments at once.

- [Deleting a Specific Developer Comment](#)
- [Deleting Multiple Developer Comments](#)

Deleting a Specific Developer Comment

To delete a developer comment:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment, Bug, or To Do** icon.
4. Click **View Comments**.
The Developer Comments page appears.
5. Locate the comment to be deleted.
6. Click the **Edit** icon.
The Edit Comment page appears.
7. Click **Delete**.

Deleting Multiple Developer Comments

To delete multiple developer comments:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment, Bug, or To Do** icon.
4. Click **View Comments**.
5. Click **Delete Comments**.
6. Select one of the following actions:
 - **Delete all comments**

- **Delete comments created by a developer**
 - **Delete comments by date**
7. Follow the on-screen instructions.

Accessing the Developer Comments from Shared Components

To access the Developer Comments report from Shared Components:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**
4. Under Tasks, click **Developer Comments**.

The Developer Comments report appears.

A navigation bar appears at the top of the page and contains the following controls:

- **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search to only specific columns. To search all columns, select **All Columns**.
- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search.
- **Actions menu** - Use the Actions menu to customize the report view. See "[About the Actions Menu](#)."



See Also:

"[Viewing and Editing Developer Comments](#)" and "[Deleting Developer Comments](#)"

Viewing the Developer Comments Calendar

To access the Developer Comments Calendar:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**
4. Under Tasks, click **Developer Comments**.

The Developer Comments page appears.

5. Click **Developer Comments Calendar**.

The Developer Comments Calendar page appears.

6. Use the Previous, Today and Next buttons to navigate.

Using the Find Icon

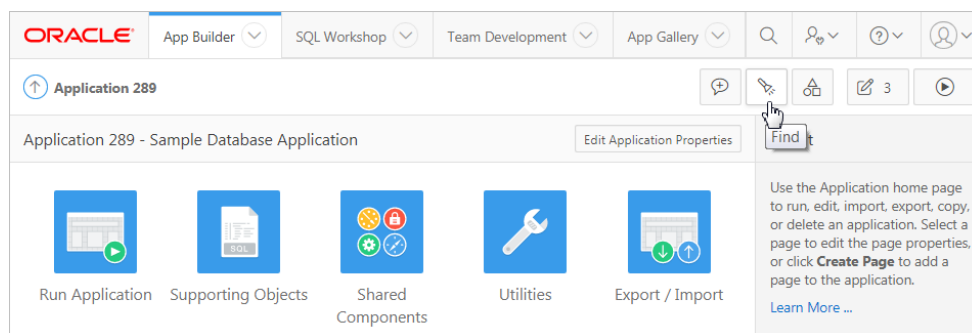
The Find icon resembles a flashlight and displays on many pages in App Builder including Application home page, Shared Components, and the Edit Application Definition. Click the Find icon to search for items, pages, queries, tables, PL/SQL code, or images, view debug reports, view session state, and view errors.

- [Accessing the Find Icon](#)
- [Searching for Items](#)
- [Searching for Pages](#)
- [Searching for Queries](#)
- [Searching for Tables](#)
- [Using the PL/SQL Finder](#)
- [Searching for Images](#)

Accessing the Find Icon

To access the Find icon:

1. Click the **Find** icon.



The Items Finder appears.

2. Click one of the following tabs:
 - **Items** - See "[Searching for Items.](#)"
 - **Pages** - See "[Searching for Pages.](#)"
 - **Queries** - See "[Searching for Queries.](#)"
 - **Tables** - See "[Searching for Tables.](#)"
 - **PL/SQL** - See "[Using the PL/SQL Finder.](#)"
 - **Images** - See "[Searching for Images.](#)"
 - **Debug** - See "[Viewing Debug Reports.](#)"
 - **Session** - See "[Viewing Session State](#)"
 - **Errors** - See View errors.

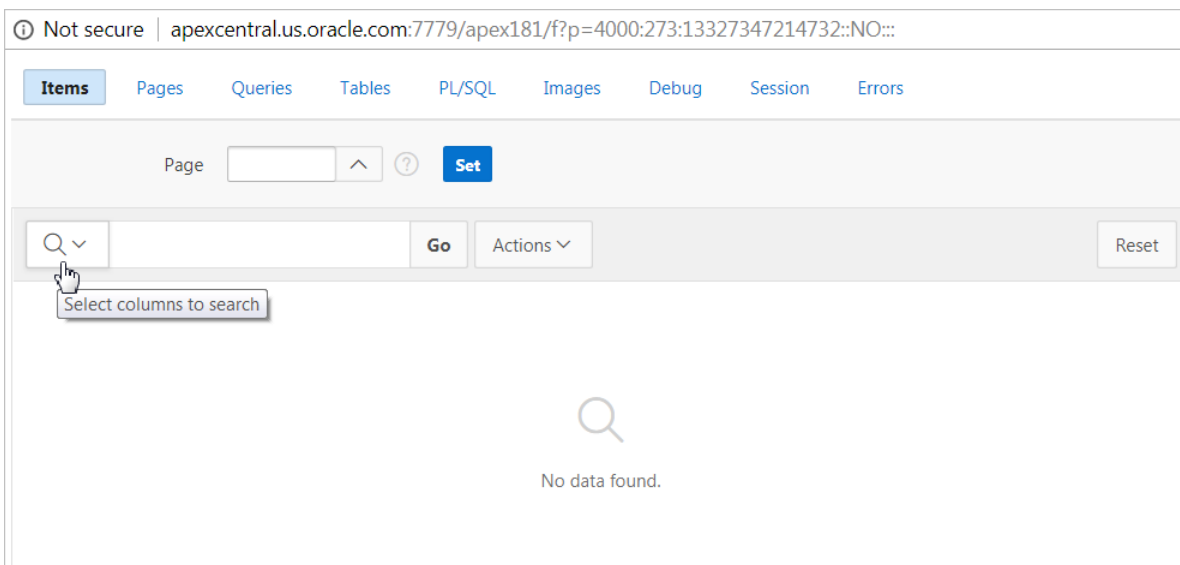
Searching for Items

In App Builder, an item can be a text field, text area, password, select list, check box, and so on. You can use the Item Finder to search for items within the current application or within the schema associated with the workspace.

To search for items:

1. Click the **Find** icon.

The Items Finder appears.



A search bar displays at the top of the page and contains the following controls:

- **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
 - **Go button** - Executes a search.
 - **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report.
2. To edit a specific item, navigate to the appropriate item.



See Also:

["Understanding Page-Level Items"](#)

Searching for Pages

A page is the basic building block of an application. You can use the Pages Finder to search for pages within the current application or within the schema associated with the workspace.

To search for a page:

1. Click the **Find** icon.
2. Click the **Pages** tab.

The Pages Finder appears. A search bar displays at the top of the page and contains the following controls:

- **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
 - **Go button** - Executes a search.
 - **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report.
3. To link to the Items page, click the page number.

Searching for Queries

You can use the Queries Finder to locate a query within your application or within the schema associated with the workspace.

To search for a query using the Queries Finder:

1. Click the **Find** icon.
2. Click the **Queries** tab.

The Queries Finder appears. A search bar displays at the top of the page and contains the following controls:

- **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
 - **Go button** - Executes a search.
 - **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report. See "[About the Actions Menu](#)."
3. To link to the Items page, click the page number.

Searching for Tables

You can use the Tables Finder to view tables within the schema associated with the workspace.

To view tables associated within the current schema:

1. Click the **Find** icon.
2. Click the **Tables** tab.

The Tables Finder appears. A search bar displays at the top of the page and contains the following controls:

- **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
 - **Go button** - Executes a search.
 - **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report.
3. Select a table name.

A table definition appears.

This report displays the column names, data type, length, precision, and scale and the SQL necessary to re-create the table appears at the bottom of the page.



See Also:

"Managing Tables" in *Oracle Application Express SQL Workshop Guide*

Using the PL/SQL Finder

You can use the PL/SQL Finder to locate and view details about stored procedures, functions, and packages associated with each object within the schema associated with the workspace.

To search for PL/SQL code in the current schema:

1. Click the **Find** icon.
2. Click the **PL/SQL** tab.

The PL/SQL Finder appears. A search bar displays at the top of the page and contains the following controls:

- **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
 - **Go button** - Executes a search.
 - **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report.
3. To view additional details, select the procedure, function, or package name.

The procedure, package, or function name appears and additional information including the owner, source name, source type, return type, argument names, data types, and IN/OUT parameters.

 **See Also:**

"Managing Database Objects with Object Browser" in *Oracle Application Express SQL Workshop Guide*

Searching for Images

You can use the Images Finder to identify images available to the current application.

To view available images:

1. Click the **Find** icon.
2. Click the **Images** tab.
The Images Finder appears.
3. From Show, select the type of images to view. Options include:
 - Standard Images
 - Workspace Images
 - Application Images

 **See Also:**

"Managing Database Objects with Object Browser" in *Oracle Application Express SQL Workshop Guide*

Adding Database Application Comments

Use Application Comments to describe an application or track developers involved in the application development.

To create an application comment:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.
4. On the Application home page, click **Shared Components**.

The Shared Components page appears.

5. Under Tasks, click **Edit Application Comment**.

The Application Comments page appears.

6. Enter comments in the Comments field and click **Apply Changes**.

Managing Application Groups

Organize applications by assigning them to application groups. To use application groups, first create a group and then assign applications to it.

- [Viewing Database Application Groups](#)
- [Creating a Database Application Group](#)
- [Assigning a Database Application to an Application Group](#)
- [Removing a Database Application from an Application Group](#)
- [Deleting an Database Application Group](#)



See Also:

["Accessing Page Specific Utilities"](#)

Viewing Database Application Groups

To view database application groups:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.

The Application Group page appears.

A Search bar displays at the top of the page. Available controls include:

- **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) to search for a group by name and click **Go**.
 - **Go button** - Executes a search or applies a filter.
 - **View Icons** (the default) displays each group as a large icon. To edit a group, click the appropriate icon.
 - **View Report** displays each group as a line in a report. To edit a group, click the name.
4. To view the applications associated with a group, you click the group name.

Creating a Database Application Group

To create a database application group:

1. On the Workspace home page, click the **App Builder** icon
2. Click **Workspace Utilities**.

3. Click **Application Groups**.
4. On the Application Group page, click **Create**.
5. Enter a name, a description (optional), and click **Create**.

Assigning a Database Application to an Application Group

To assign a database application to a group:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.
4. On the Tasks list, click **Manage Unassigned**.
The Manage Application Group Assignments page appears.
5. From the New Group list, select a group.
6. Select the applications to be assigned and click **Assigned Checked**.

Tip:

You can also assign an application to an application group by editing the application definition. Select the group from the Application Group list. See "[Editing Application Attributes](#)."

Removing a Database Application from an Application Group

To remove an application from an application group:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.
4. On the Tasks list, click **Manage Assignments**.
The Manage Application Group Assignments page appears.
5. From the New Group list, select **Unassign**.
6. Select the pages to be unassigned and click **Assigned Checked**.

Tip:

You can also remove an application from an application group by editing the application definition. Select **Unassigned** from the Application Group list. See "[Editing Application Attributes](#)."

Deleting an Database Application Group

To delete an application group:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.
4. Assign all applications in the group to Unassign:
 - a. On the Tasks list, click **Manage Assignments**.
 - b. From the New Group list, select **Unassign**.
 - c. Select the pages to be unassigned and click **Assigned Checked**.
5. Return to the Application Groups page.
6. Select the application group to be deleted.
7. On the Application Group page, click **Delete**.

A confirmation page appears.
8. Confirm your request.

Using Application Utilities

Access the Utilities page to view useful application summaries and reports.

- [Application Utilities](#)
Learn about the links available on the application Utilities page.
- [Accessing the Utilities Page](#)
Access the Utilities page by selecting the application and then clicking the Utilities icon.
- [Viewing the Application Dashboard](#)
View a report that summarizes application components and attributes.
- [Upgrading an Application to Include New Components](#)
Upgrade an existing application to include new components available in the current release.
- [Viewing the Recently Updated Pages Report](#)
Displays a report of pages recently updated by the current user.
- [Viewing the Change History Report](#)
The Change History report displays a summary report of edits to the current applications by developer, component type, and page number
- [Using the Database Object Dependencies Report](#)
The Database Object Dependencies report identifies database objects referenced by the current application. Review this report to determine what objects to move when deploying an application.
- [Accessing Application Express Views](#)
Query various views against Application Express metadata.
- [Accessing Page Specific Utilities](#)
Use the links in the Page Specific Utilities region to access Cross Page Utilities and Page Groups as well as access handy utilities for regions, buttons, items, computations, validations, process, dynamic actions, and branches.
- [Accessing Cross Page Utilities](#)
Use Cross Page Utilities to perform edits across pages in the application.

- [Using Interactive Report Utilities](#)
View and delete saved interactive reports and interactive report subscriptions.



See Also:

"[Creating Custom Activity Reports Using APEX_ACTIVITY_LOG](#)"

Application Utilities

Learn about the links available on the application Utilities page.

The following table describes the links on the application Utilities page.

Table 6-21 Links on the Application Utilities Page

| Link | Description | To Learn More |
|------------------------------|---|---|
| Application Dashboard | View a summary of application components and attributes | See " Viewing the Application Dashboard " |
| Advisor | Performs various checks on your application, including checks for programming errors, security issues, quality assurance, and other best practice | See: " Running Advisor to Check Application Integrity " |
| Upgrade Application | Upgrade the current application to a new release. Review components eligible for upgrading to include the latest features. | See " Running Upgrade Application " |
| Recently Updated Pages | View a history of page updates made to this application. | See " Viewing the Recently Updated Pages Report " |
| Attribute Dictionary | Manage item and column user interface defaults for a selected page. | See " Using the Attribute Dictionary " |
| Change History | View a report of component updates made in this application. | See " Viewing the Change History Report " |
| Database Object Dependencies | View a report of database object referenced by this application. | See " Using the Database Object Dependencies Report " |
| Debug Messages | Review debug messages generated by this application. | See " Debugging an Application " |
| Application Express Views | Query various views against Application Express metadata. | See " Accessing Application Express Views " |
| Export Repository | View a history of application exports. | See " Accessing the Export Repository " |

Table 6-21 (Cont.) Links on the Application Utilities Page

| Link | Description | To Learn More |
|------------------------------|---|--|
| Page Specific Utilities | Page Specific Utilities display in a region on the right-side of the Utilities page. Use the links in this region to access Cross Page Utilities and Page Groups as well as access handy utilities for managing regions, buttons, items, computations, validations, process, dynamic actions, and branches. | See: <ul style="list-style-type: none"> • "Accessing Page Specific Utilities" • "Accessing Cross Page Utilities" |
| Interactive Report Utilities | Interactive Report Utilities display in a region on the lower right-side of the Utilities page. Click Saved Reports to delete saved interactive reports in the current application. Click Subscriptions to manage interactive report subscriptions. | See "Using Interactive Report Utilities" |

Accessing the Utilities Page

Access the Utilities page by selecting the application and then clicking the Utilities icon.

To access App Builder Utilities page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.

The Utilities page features links to the following tools and reports:

- **Application Dashboard**
- **Advisor**
- **Upgrade Application**
- **Recently Updated Pages**
- **Attribute Dictionary**
- **Change History**
- **Database Object Dependencies**
- **Debug Messages**
- **Application Express Views**
- **Export Repository**

 **See Also:**

- ["Application Utilities"](#)
- ["Accessing Page Specific Utilities"](#)

Viewing the Application Dashboard

View a report that summarizes application components and attributes.

To view the Application Dashboard:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

The Application home page appears.

4. Click **Utilities**.
5. Click **Application Dashboard**.

The Application Dashboard appears and contains the following sections:

- **Application Overview** lists detailed information and statistics about the current application, including the ID, name, alias, parsing schema, associated group, number of pages, and associated theme.
 - **Security** lists the current authentication scheme, number of public and non public pages, and the number of authorization schemes used within the current application.
 - **Templates** contains links to reports of templates used within the current application.
 - **Pages by Type** lists counts of components, including types of reports and forms and dynamic HTML.
 - **Application Components** contains links to reports of application-level controls and logic, including lists, shortcuts, lists of value, tabs, application computations, application items, plug-ins, and application processes.
 - **Page Components** contains links to reports of page-level controls and logic, including regions, items, branches, computations, validations, dynamic actions, classic report columns, interactive report columns, and button.
 - **Native Region Component Utilization** charts the total number of all types of region components used in the app. This is a visualization of the information provided in the Pages by Type section, the Page Components, and others.
 - **Plugin Region Utilization** charts the total number of plug-ins used in the app.
6. To view a specific report, click the number link to the right of the component type or template.



See Also:

["Identifying At Risk Password Items"](#)

Upgrading an Application to Include New Components

Upgrade an existing application to include new components available in the current release.

- [About Upgrading to Include New Components](#)
- [Running Upgrade Application](#)

About Upgrading to Include New Components

The Upgrade Application page displays the available upgrade types along with the number of objects that can be upgraded. Click on the number under Candidate Object to review the candidates and upgrade.

Running Upgrade Application

To upgrade an application to use new components:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.
The Application home page appears.
4. Click **Utilities**.
5. Click **Upgrade Application**.
The available upgrade types display along with the number of objects that can be upgraded.
6. To review the candidates and upgrade, click the number link under **Candidate Objects**.
7. Follow the on-screen instructions.

Viewing the Recently Updated Pages Report

Displays a report of pages recently updated by the current user.

To view the Recently Updated Pages report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.
The Application home page appears.
4. Click **Utilities**.

5. Click **Recently Updated Pages**.

The Recently Updated Pages report appears.

Viewing the Change History Report

The Change History report displays a summary report of edits to the current applications by developer, component type, and page number

To view the Change History report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

The Application home page appears.

4. Click **Utilities**.
5. Click **Change History**.

The Change History report appears. Use the **Since** list to control the amount of time to include in your report.

6. From **Since** at the top of the page, select the amount of time to include in your report and click **Set**.

Using the Database Object Dependencies Report

The Database Object Dependencies report identifies database objects referenced by the current application. Review this report to determine what objects to move when deploying an application.

To view the Database Object Dependencies report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

The Application home page appears.

4. Click **Utilities**.
5. Click **Database Object Dependencies**.

The Database Object Dependencies page appears.

6. Click **Compute Dependencies**.

Tip:

Display of the report may take a moment depending on the size and complexity of your application.

The Database Object Dependencies report appears.

7. To view the components that reference a specific database object, click the number under **Reference Count**.

Accessing Application Express Views

Query various views against Application Express metadata.

Application Express Views are data dictionary views that expose the metadata for applications. In Application Express Views, you cannot only see the view, but you can also see the data they contain.

To access the Application Express Views page:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

The Application home page appears.

4. Click **Utilities**.
5. Click **Application Express Views**.

The Application Express Views report appears. Use the Search bar at the top of the page to customize the display.

Tip:

Click **Tree View** to switch to a relational view of this information with expandable parent and children entries.

6. Select a view.
7. On Selected Columns, select columns from **Select Columns** and click the **Results** button.
The Results report appears.
8. On Results:
 - To view the query that was executed, expand the **Query** region . You can copy this query for reuse with SQL Commands or SQL Developer.
 - Click **Download**, to download to a file.

Accessing Page Specific Utilities

Use the links in the Page Specific Utilities region to access Cross Page Utilities and Page Groups as well as access handy utilities for regions, buttons, items, computations, validations, process, dynamic actions, and branches.

To access Page Specific Utilities page:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.

The App Builder home page appears

3. Select an application.
The Application home page appears.
4. Click **Utilities**.
5. Under Page Specific Utilities, select one of the following:
 - Cross Page Utilities
 - Page Groups
 - Region Utilities
 - Button Utilities
 - Item Utilities
 - Computations Utilities
 - Validation Utilities
 - Process Utilities
 - Dynamic Action Utilities
 - Branch Utilities



See Also:

["Accessing Cross Page Utilities"](#)

Accessing Cross Page Utilities

Use Cross Page Utilities to perform edits across pages in the application.

To access the Cross Page Utilities page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities, click **Cross Page Utilities**.

The Cross Page Utilities page provides access to the following tools and reports:

- **Grid Edit of all Pages.** Use this report to edit the page name, alias, title, page template, and security options. To link to a specific page, click the Edit icon adjacent to the page number. To save your changes, click **Apply Changes**.
- **Delete Multiple Pages.** Delete multiple pages at once.
- **Page Attributes.** View a report of details for all pages in the current application including the page name, page title, defined groups, type and number of components and controls, and application logic. To customize the report, use the search bar at the top of the page. To link to a specific page, click the page number.
- **Page Locks.** Use Page Locks to prevent conflicts during application development. By locking a page, you prevent other developers from editing it. See and

 **See Also:**

- ["Locking and Unlocking a Page"](#)
- ["Deleting Multiple Pages"](#)
- ["Using Application Utilities"](#)

Using Interactive Report Utilities

View and delete saved interactive reports and interactive report subscriptions.

- [Deleting Saved Interactive Reports](#)
View and delete saved interactive reports in the current application.
- [Viewing Subscriptions](#)
View and delete interactive report subscriptions.

Deleting Saved Interactive Reports

View and delete saved interactive reports in the current application.

To delete saved interactive reports:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. Click **Utilities**.
4. Under Interactive Report Utilities, select **Saved Reports**.
The report displays all of the saved report for the application which are not primary defaults. Each row includes the page number, region name, report type, report name, report alias, and owner.
5. To view private saved reports, deselect the checkbox for the **Primary Default** filter located below the search bar.
6. To delete a report, select the check box adjacent to the page number and click **Delete Checked**.
7. To view the page containing the report, click the page number.

 **See Also:**

- ["Linking to Shared Interactive Reports"](#)
- ["Saving Interactive Reports"](#)
- "Managing Saved Interactive Reports" in *Oracle Application Express Administration Guide*

Viewing Subscriptions

View and delete interactive report subscriptions.

End users can receive updated versions of a report by subscribing to it. Workspace administrators can view and manage these subscriptions on the Subscriptions page.

To manage interactive report subscriptions:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
The Application home page appears.
3. Click **Utilities**.
4. Under Interactive Report Utilities, select **Subscriptions**.
The Subscriptions report appears.
5. To delete a subscriptions, select the check box adjacent to the page number and click **Delete Checked**.

7

Managing Pages in a Database Application

A page is the basic building block of an application. Developers add pages to an application by running the Create Page Wizard.

- [About Creating New Pages](#)
Developers add pages to an application by running the Create Page Wizard.
- [Adding a New Page to an Application](#)
Add a new page by running the Create Page Wizard from the Application home page.
- [Understanding Page Types in the Create Page Wizard](#)
Run the Create Page Wizard to add a new page to existing application. How the wizard works depends upon the type of page you are creating.
- [Creating Dialog Pages](#)
Oracle Application Express includes support for modal and non-modal dialog pages. Modal and non-modal dialog pages are only supported if the current application theme contains at least one page template of template type Dialog Page.
- [About Copying Database Application Pages](#)
Copy a page from the current application or from another application.
- [About Deleting Database Application Pages](#)
Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries. You can only delete unlocked pages or pages that you have locked.
- [Creating Page Groups](#)
Organize pages into page groups to make them access easier. To use page groups, you create a group and then assign pages to the group.
- [Locking and Unlocking a Page](#)
Prevent conflicts during application development by locking pages in your application. By locking a page, you prevent other developers from editing it.



See Also:

["Editing Pages in Page Designer"](#) and ["Using Application Utilities"](#)

About Creating New Pages

Developers add pages to an application by running the Create Page Wizard.

The Create Page Wizard includes two types of pages:

- **Component** pages provide page-level functionality and can only be added once within a given application. Available Component pages include Report, Form,

Plug-ins, Chart, Calendar, Tree, Wizard, Data Loading, Legacy Page, and Dashboard Page.

- **Feature** pages provide application-level functionality and can only be added once per application. Available Features include About Page, Access Control, Activity Reporting, Configuration Options, Email Reporting, Feedback, Login Page, and Theme Style Selection.

You can access the Create Page Wizard as follows:

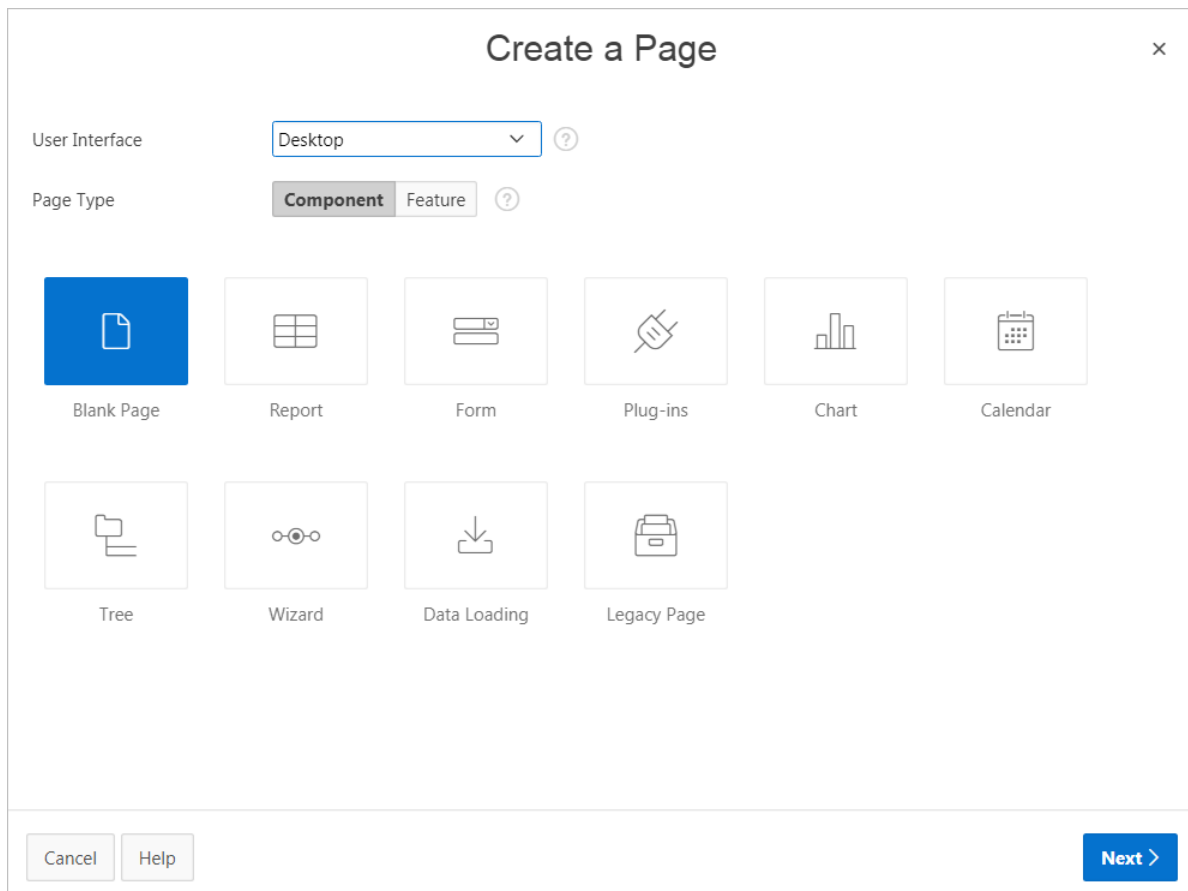
- Application home page — Click **Create Page**. See "[Adding a New Page to an Application](#)."
- Viewing a page in Page Designer — Click **Create** on the Page Designer Developer toolbar and select **Page**. See "[Page Designer Toolbar](#)"

Adding a New Page to an Application

Add a new page by running the Create Page Wizard from the Application home page.

To create a new page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
The Application home page appears.
3. Click the **Create Page** button.
The Create Page Wizard appears.



4. For Create a Page:

- a. User Interface - Select a user interface for the page.

This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.

- b. Page Type - Select the type of page you want to create. Options include:

- **Component** - Provides page-level functionality and can be added multiple times within a given application. Available Component pages include Blank Page, Report, Form, Plug-ins, Chart, Calendar, Tree, Wizard, Data Loading, Legacy Page, and Dashboard Page.
- **Feature** - Provides application-level functionality. Available Features pages include About Page, Access Control, Activity Reporting, Configuration Options, Email Reporting, Feedback, Login Page, and Theme Style Selection.

 **Tip:**

The Page Type you select(that is, **Component** or **Feature**) determines what appears next. To learn more see, "[Understanding Page Types in the Create Page Wizard.](#)"

- c. Click **Next**.

5. Follow the on-screen instructions.



See Also:

["About Creating New Pages"](#)

Understanding Page Types in the Create Page Wizard

Run the Create Page Wizard to add a new page to existing application. How the wizard works depends upon the type of page your are creating.



Tip:

Universal Theme - 42 is optimized to work equally well in either a mobile or desktop environment. In previous releases and in earlier themes, the Mobile user interface is based on jQuery Mobile. Oracle recommends migrating existing mobile applications to the Universal Theme as soon as possible.

- [Available Component Page Types](#)
Learn about available components when running the Create Page Wizard using the Universal Theme.
- [Available Features Page Types](#)
Learn about available features when running the Create Page Wizard using the Universal Theme.
- [Available Components for Mobile User Interface Applications](#)
Learn about available components when running the Create Page Wizard using the Mobile User Interface.
- [About Removing Features Created with the Create Page Wizard](#)
Remove features created with a wizard using build options.

Available Component Page Types

Learn about available components when running the Create Page Wizard using the Universal Theme.

Table 7-1 Desktop Applications — Component Types

| Page Type | Description | To Learn More |
|------------|--|--|
| Blank Page | Creates a page with no built-in functionality. | See online Help when creating this page. |

Table 7-1 (Cont.) Desktop Applications — Component Types

| Page Type | Description | To Learn More |
|---------------|---|---|
| Report | A report is the formatted result of a SQL query. Report options include: <ul style="list-style-type: none"> Interactive Report Interactive Grid Classic Report Report and Form on Table List View (Optimized for mobile) Column toggle Report (Optimized for mobile) Reflow Report (Optimized for mobile) Report on Web Service | See: <ul style="list-style-type: none"> "Developing Reports" "Understanding Report Types" "Report Options When Running the Create Page Wizard" |
| Form | Forms enable users to update a single row or multiple rows in a table. Form options include: <ul style="list-style-type: none"> Report with Form on Table Report with List View on Table (Optimized for mobile) Editable Interactive Grid Form on a Table Form on a Procedure Form on a SQL Query Form on a Web Service Report and Form on a Web Service | See: <ul style="list-style-type: none"> "Creating a Form" "Understanding Form Types" "Form Options When Running the Create Page Wizard" |
| Master Detail | Create a master detail form that enables users to query, insert, update, and delete values from related tables or views. You choose the tables on which to build the master and detail regions. Master Detail options include: <ul style="list-style-type: none"> Stacked - Creates a single page master detail with editable interactive grids. Side by Side - Creates a single page (or Side by Side) master detail utilizing side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report. Drill Down - Creates a two page (or Drill Down) master detail. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grids for the detail. | See: <ul style="list-style-type: none"> "Managing Master Detail Forms" "About Master Detail Forms" |
| Plug-ins | Creates a new page based on a region type plug-in. Plug-ins enable developers to declaratively extend, share, and reuse the built-in types available with Oracle Application Express. | See "Importing a Plug-in from the Plug-in Page" |
| Chart | Enables you to create graphical charts. | See "Creating Charts" |
| Dashboard | Creates a first-cut dashboard based on sample data that you can easily customize using Page Designer. Choose from a number of different chart layouts as the starting point for your page. | See online Help when creating this page. |

Table 7-1 (Cont.) Desktop Applications — Component Types

| Page Type | Description | To Learn More |
|--------------|---|--|
| Calendar | Generates a calendar with monthly, weekly, and daily views. | See " Creating Calendars " |
| Tree | Creates a tree to graphically communicate hierarchical or multiple level data and optionally navigating to a page or URL specific to each tree node. | See " Managing Trees " |
| Wizard | Create a collection of pages to be used as a wizard. Generally wizards are used for entering data across multiple steps. | See online Help when creating this page. |
| Data Loading | Creates a Data Loading wizard allowing the end user to manage the loading of data into a table to all schemas for which the user has privileges. | See " Creating Applications with Data Loading Capability " |
| Legacy Page | Legacy pages contains components Oracle still supports but does not recommend using. Options include: <ul style="list-style-type: none"> • Tabular Form • Legacy Master Detail • AnyChart Chart • Summary Page • Legacy Calendar | See " Managing Application Legacy Components " |

Available Features Page Types

Learn about available features when running the Create Page Wizard using the Universal Theme.

This table lists available Features for Desktop Applications when running the Create Page Wizard.

Table 7-2 Create Page Wizard — Features

| Page Type | Description | To Learn More |
|----------------|---|--|
| About Page | Include an About this Application page which features a description field that describes the application, includes the application version, and a count of the number of pages | See online Help when creating feature. |
| Access Control | Creates pages to manage an access control list. Use the Application Access Control shared component to associate application roles with application users. This wizard also adds a reader , contributor and administrator role and corresponding authorization scheme to your application. Apply these authorization schemes to pages and page components to manage access by user and role. | See " Controlling Access to Applications, Pages, and Page Components " |

Table 7-2 (Cont.) Create Page Wizard — Features

| Page Type | Description | To Learn More |
|-----------------------|--|--|
| Activity Monitoring | Include numerous reports on end user activity for your application. Determine the most active users, the most used pages, the performance of pages, and errors raised, to better understand how your application is being utilized and areas for improvement. | Not applicable. |
| Configuration Options | Enables application administrators to enable or disable specific functionality within the application. This feature is useful if you select features that need additional development effort before they can be used by end users. You can expand this feature to application-specific features. If developers define additional build options and associate them with specific functionality throughout the application, then they can be added to the configuration settings for administrators. | Not applicable. |
| Email Reporting | Include numerous reports on emails queued from this application, emails sent, and errors sending emails. | Not applicable. |
| Feedback | Feedback provides a mechanism for end users to post general comments for application administrators and developers. The posts include useful session state information to help developers determine where the end user sent the feedback from. Creating Feedback: <ul style="list-style-type: none"> • Creates Navigation bar icon which users can click to leave feedback. • Creates an report for viewing and updating feedback. • Captures the application and page ID, feedback comments, date and time, and user information. | See " Managing Feedback ". |
| Job Reporting | This option only appears if you have database jobs associated with the underlying schemas for the workspace. Include reports on jobs and job run details within the schema associated with this application. | Not applicable. |
| Login Page | Creates a Login Page which enables the user to enter login credentials for accessing your application. Developers can select a template and labels for the Username and Password fields as well as determine if the Username value is stored a system-generated cookies. | Not applicable. |

Table 7-2 (Cont.) Create Page Wizard — Features

| Page Type | Description | To Learn More |
|-----------------------|--|-----------------|
| Theme Style Selection | Enables administrators to select a default color scheme (theme style) for the application. Administrators can also choose to allow end users to choose their own theme style. End users can simply click on the Customize button at the bottom of the home page, and select from the available theme styles. For example, users with visual impairment may prefer to utilize the Vista theme style which has a much higher color contrast. | Not applicable. |



See Also:

["About Removing Features Created with the Create Page Wizard"](#)

Available Components for Mobile User Interface Applications

Learn about available components when running the Create Page Wizard using the Mobile User Interface.

The Mobile User Interface based on jQuery Mobile has been desupported in Application Express 18.1. Only apps built in previous releases can have an associated Mobile User Interface.



Tip:

Universal Theme - 42 is optimized to work equally well in either a mobile or desktop environment. In previous releases and in earlier themes, the Mobile user interface is based on jQuery Mobile. Oracle recommends migrating existing mobile applications to the Universal Theme as soon as possible.

In previous releases, developers selected the **Mobile** User Interface to optimize applications for mobile environments. The Mobile user interface is based on jQuery Mobile. If your application uses an older theme and the Mobile User Interface, the Create Page Wizard supports the following page types.

Table 7-3 Mobile User Interface Applications — Component Types

| Page Type | Description | To Learn More |
|------------|--|----------------|
| Blank Page | Creates a page with no built-in functionality. | Not applicable |

Table 7-3 (Cont.) Mobile User Interface Applications — Component Types

| Page Type | Description | To Learn More |
|--------------|---|---|
| Report | A report is the formatted result of a SQL query. Available report options include: <ul style="list-style-type: none"> List View Column Toggle Report Reflow Report Report on Web Service | See: <ul style="list-style-type: none"> "Developing Reports" "Understanding Report Types" "Report Options When Running the Create Page Wizard" |
| Form | Forms enable users to update a single row or multiple rows in a table. Available form options include: <ul style="list-style-type: none"> Report with List View on Table Form on a Table Form on a Procedure Form on a SQL Query Form on a Web Service Report and Form on a Web Service | See: <ul style="list-style-type: none"> "Creating a Form" "Understanding Form Types" "Form Options When Running the Create Page Wizard" |
| Plug-ins | Creates a new page based on a region type plug-in. Plug-ins enable developers to declaratively extend, share, and reuse the built-in types available with Oracle Application Express. | See "Importing a Plug-in from the Plug-in Page" |
| Chart | Enables you to create graphical charts. | See "Managing Legacy Charts" |
| Calendar | Generates a calendar with monthly, weekly, and daily views. | See "Creating Calendars" |
| Tree | Creates a tree to graphically communicate hierarchical or multiple level data and optionally navigating to a page or URL specific to each tree node. | See "Managing Trees" |
| Wizard | Create a collection of pages to be used as a wizard. Generally wizards are used for entering data across multiple steps. | See online Help when creating this page. |
| Data Loading | Creates a Data Loading wizard allowing the end user to manage the loading of data into a table to all schemas for which the user has privileges. | See "Creating Applications with Data Loading Capability" |
| Legacy Page | Legacy pages contains components Oracle still supports but does not recommend using. Options include: <ul style="list-style-type: none"> Tabular Form Legacy Master Detail AnyChart Chart Summary Page Legacy Calendar | See "Managing Application Legacy Components" |



See Also:

["Creating Applications for Mobile Devices"](#)

About Removing Features Created with the Create Page Wizard

Remove features created with a wizard using build options.

When you add a feature using a wizard, the wizard creates one or more pages and other components and processes to seamlessly integrate it into your application. Each feature is associated with a build option which contains one or more components. You use the associated build option to enable, disable, or permanently remove features.

Using Build Options to Include or Exclude Features

Build options have two possible values: **Include** and **Exclude**. If you select the build option status of **Include**, then the Application Express engine considers the associated components (in this case features) as part of the application definition at runtime. Conversely, if you specify the build options status as **Exclude**, then the Application Express engine treats it and any associated components as if it did not exist.

Removing Features Permanently

You remove features created with a wizard by first removing the components associated with the build option and then deleting the build option.



See Also:

- ["Including or Excluding Build Options"](#)
- ["Deleting Build Options and Associated Components"](#)

Creating Dialog Pages

Oracle Application Express includes support for modal and non-modal dialog pages. Modal and non-modal dialog pages are only supported if the current application theme contains at least one page template of template type Dialog Page.

- [About Supported Dialogs](#)
- [Using a Dialog Page Template](#)
- [Defining a Dialog Page as a in Page Designer](#)
- [Generating a Cancel Dialog Dynamic Action on the Dialog Page](#)
- [Generating a Close Dialog Page Process on the Dialog Page](#)
- [About Branching with Dialog Pages](#)
- [Handling Refresh Actions on Parent Page](#)

About Supported Dialogs

When you create a new page, the wizard prompts you to select a Page Mode. The Page Mode determines if the page is a **normal** application page or a **dialog** page. Oracle Application Express supports two types of dialog pages:

- **Modal Dialog** - A modal dialog is an overlay window positioned within the within the same browser window. A modal dialog remains active and focused until the user has finished with it and closes it. While a modal dialog is active, the user is unable to interact with the rest of the page until the dialog is closed.
- **Non-modal Dialog** - A non-modal dialog displays a separate popup browser window. A user can interact with a non-modal dialog and content on the page. Use a non-modal dialog when the requested information is not essential to continue. This type of window can be left open while work continues elsewhere.

Using a Dialog Page Template

Oracle Application Express includes dialog page templates in the *Universal Theme - 42*. When you set the Page Mode to either **Modal** or **Non-Modal**, Oracle Application Express automatically uses the default dialog page template. This section describes how to create a dialog template if your theme does not include one and how to edit dialog template attributes.

- [Creating a Dialog Page Template](#)
- [About Defining Dialog Page Template Attributes](#)

Creating a Dialog Page Template

If you are using an existing application and the theme does not contain such a dialog page template, then you must create one. While you can create one from scratch, the easiest option is to copy an existing dialog template.

To create a dialog page template:

1. Copy the dialog page template from the Sample Database application:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select the application to which you want to copy the template.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Templates**.
 - e. Click **Create**.
 - f. In the Create Template Wizard:
 - Template Type - Select **Page** and click **Next**.
 - Creation Method - Select **As a Copy of an Existing Template** and click **Next**.
 - Copy From - Select **Sample Database Application** and click **Next**.
 - Identify Theme - Select the theme you are copying from and the theme you are copying to and click **Next**.
 - New Template - Locate the Modal Dialog template. Under Copy, select **Yes**.
 - Select Create **Copy Page Templates**.
2. Verify the dialog page template is active in the current theme. See "[Selecting a Default Page Template](#)."

About Defining Dialog Page Template Attributes

The dialog page template contains JavaScript function calls that handle the initialization, closure, and cancellation of a dialog page. When you create a dialog page template you must define these and other Dialog Page Templates attributes.



See Also:

["Required Dialog Page Template Attributes"](#)

Defining a Dialog Page as a in Page Designer

To define a page as a dialog in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In Page Designer locate the Appearance.
3. For Page Mode, select one of the following:
 - **Modal Dialog** - The page is presented as a modal dialog. A modal dialog is an overlay window positioned within the viewport, which remains active and focused until the end user dismisses (closes) it. The underlying page is grayed out and the end user is prevented from interacting with the rest of the page until the dialog is closed.
 - **Non-Modal Dialog** - The page is presented as a non-modal dialog. A non-modal dialog is an overlay window positioned within the viewport, where the end user can interact with the dialog, and content on the page that launched the dialog. A non-modal dialog can be used when the requested information is not essential to continue, work can continue elsewhere while the window is still open.
4. Edit other Appearance attributes as needed.



Tip:

To learn more about an attribute, see field-level Help.

5. Under Dialog, edit the appropriate attributes:
 - a. **Width** - Enter the width of the dialog, in pixels or a percentage. The substitution string `#DIALOG_WIDTH#` in the dialog initialization code of the page template is replaced with the value entered in this item. For a page using a Desktop user interface, the value is set in pixels, such as 500. However, for a

page using the jQuery Mobile Smartphone user interface, the value is set in percent, such as 80%.

- b. **Height** - Enter the height of the dialog, in pixels, for example, 500. The substitution string `#DIALOG_HEIGHT#` in the dialog initialization code of the page template is replaced with the value entered.
 - c. **Maximum Width** - Enter the maximum width of the dialog in pixels. The substitution string `#DIALOG_MAX_WIDTH#` in the dialog initialization code of the page template is replaced with the value entered in this item.
 - d. **Attributes** - Dialog support varies depending on the user interface of the page. A subset of dialog attributes are declaratively supported: height, width, and maximum width. To define any additional attributes for the dialog, enter the name/value pairings in a comma separated list. The substitution string `#DIALOG_ATTRIBUTES#` in the dialog initialization code defined at page template level, is replaced by any value(s) entered in this item.
 - e. **CSS Classes** - Enter classes to add to this component. You may add multiple classes by separating them with spaces.
 - f. **Chained** - Select **Yes** if this dialog is opened by another dialog, and should reuse the same dialog window. For example, if this dialog is part of a multiple page dialog wizard and you wish to reuse the dialog window for each step of the wizard. Select **No** if this dialog is opened by another dialog and should be opened in a new dialog window. For example, when this dialog is modal, and you wish to open it within a parent modal dialog window.
6. To save your changes, click **Save**.

**See Also:**

["About Creating New Pages"](#)

Generating a Cancel Dialog Dynamic Action on the Dialog Page

If your dialog page contains a Cancel button, then you must generate a dynamic action based upon the Click of that button, with its True Action event being set to **Cancel Dialog**. This dynamic action will use the cancellation JavaScript call contained in the page template (`apex.navigation.dialog.cancel`) to handle the cancelling the dialog.

**See Also:**

["Managing Dynamic Actions"](#)

Generating a Close Dialog Page Process on the Dialog Page

You close a dialog page in Oracle Application Express by creating a Close Dialog page process. In previous releases, this page process was referred to as a Close Popup Window. This dialog will use the closure JavaScript call contained in the page

template (`apex.navigation.dialog.close`), to handle the closing of the dialog. The page process also supports the return of dialog page items, which you may wish to reference on the parent page. To view examples, edit any of the Dialog pages in the *Sample Database Application*.

 **Tip:**

You can also close a dialog page with the Close Dialog dynamic action. Both the dynamic action and the process support returning dialog page items. Use the page process when the page needs to be submitted and use the dynamic action when the page does not need to be submitted. Both end up calling the closure JavaScript call. Alternatively, you can create a page branch, which will close a dialog page and redirect to the page specified in the branch.

 **See Also:**

- ["Understanding Page Processes"](#)
- ["Understanding Sample Database Application"](#)
- ["Controlling Navigation Using Branches"](#)

About Branching with Dialog Pages

You can open dialog pages using buttons or links. Dialog pages support the following branching options:

- Dialog page to dialog page.

You can branch from one dialog page to another dialog page if:

- Both pages have the same Page Mode, for example Modal Dialog and the dialog Chained attribute is set to **Yes**. Setting the Chained attribute to **Yes** enables support for chained or nested dialogs such as the dialogs that appear many internal Application Express wizards. Chained or nested dialogs enable you to re-use a dialog window to display each page.
- Dialog page to Normal page.
 - When branching from a dialog page to a normal page, the dialog page is closed and the user navigates to a new page in the browser. The new page can be parent (or launching) page or another Normal page in an application.

Handling Refresh Actions on Parent Page

If you have generated a Close Dialog page process on the dialog page, you must implement refresh actions on the parent page using the new `Dialog Closed` dynamic action event type. You can use the set type Dialog Return Item when creating a Set Value True Action, to set a single value, based on the return item of a dialog.

To see an example, go to the Products page in the *Sample Database Application*. On the Products page, the Create Product button has an associated dynamic action that launches the dialog. Closing the dialog triggers the `Dialog_Closed` dynamic action. You can use this type of dynamic action for whatever True Actions you wish to define. In this example, it refreshes the report on the parent page and submits the page.

 **Tip:**

For a partial page refresh of a parent page, implement the close dialog with dynamic action (for example, for report and form pages where you want to edit data in a modal dialog form and only want the report refreshed on close). In contrast, branching is the best approach if you have a multi step modal wizard or if you have several possible target pages when closing the dialog.

 **See Also:**

- ["Installing and Running Sample Database Application"](#)
- ["Managing Dynamic Actions"](#)
- ["Understanding Sample Database Application"](#)

About Copying Database Application Pages

Copy a page from the current application or from another application.

During the copy process, you can also copy shared components or change mappings to shared components in the target application.

About Copying Pages Between User Interfaces in Older Themes

The Copy Page Wizard enables you to copy a page to a different user interface. During the copying process, the user can select the user interface for the new page. If the application contains a Desktop and a Mobile theme, both user interface options display.

 **Note:**

If your application is using an older theme, Oracle does not support copying Desktop pages to jQuery Mobile Smartphone pages or copying jQuery Mobile Smartphone pages to Desktop pages. Attempting to copy the following Desktop pages to a Mobile user interface results in an error:

- Interactive report
- Flash map
- Flash chart
- Tree region

 **See Also:**

["Copying a Database Application Page "](#)

About Deleting Database Application Pages

Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries. You can only delete unlocked pages or pages that you have locked.

 **See Also:**

["Deleting Pages"](#)

Creating Page Groups

Organize pages into page groups to make them access easier. To use page groups, you create a group and then assign pages to the group.

 **Tip:**

Page groups do not have any function other than to aid developers in organizing their application pages.

- [Viewing Page Groups](#)
- [Creating a Page Group](#)
- [Editing a Page Group Definition](#)
- [Assigning a Page to a Page Group](#)

- [Viewing Pages by Page Group](#)
- [Deleting a Page Group](#)

Viewing Page Groups

To access the Page Groups page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.

The Page Groups page appears.

A Search bar displays at the top of the page. Available controls include:

- **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) to search for a page group by name and click **Go**.
 - **Go button** - Executes a search or applies a filter.
 - **View Icons** - Displays each group as a large icon. To edit a group, click the appropriate icon.
 - **View Report** - Displays each group as a line in a report. To edit a group, click the name.
 - **Actions menu** - Use the Actions menu to customize the report view.
5. To view the pages associated with a group, click **Pages by Page Group**.



See Also:

["About the Actions Menu"](#)

Creating a Page Group

To create a page group:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. On the Page Groups page, click **Create**.
6. Enter a name, a description (optional), and click **Create**.

Editing a Page Group Definition

When you create a page group you specify a name and description.

To edit the Page Group definition:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. Select a page group.
6. On the Page Group page, edit the name or description and click **Apply Changes**.
7. To move to the next Page Group definition, click the **Previous** (<) and **Next** (>) buttons and repeat the previous step.

Assigning a Page to a Page Group

To assign pages to a group:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. Click the **Page Assignments** tab.
The Page Assignments page appears.
6. Click the page number.
The Page Attributes page appears.
7. Locate the Name region and select a group from the Group list.
8. Click **Apply Changes**.

Viewing Pages by Page Group

To view pages by page group:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. Click the **Pages by Page Group** tab.
The Pages by Page Group page appears. Click the Page Name to view the page.
6. To remove a page from a group:
 - a. Click the page number.
The Page Attributes page appears.

- b. Under Name, Group, select `- No Group Assigned -`.
 - c. Click **Apply Changes**.
7. To create a new group, click **Create** and follow the on-screen instructions.

Deleting a Page Group

To delete a page group:

1. Remove all pages from the group to be deleted:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select the application.
 - c. From Page Specific Utilities, click **Page Groups**.
 - d. Click the **Pages by Page Group** tab.
 - e. Click the page number.

The Page Attributes page appears.
 - f. Under Name, Group, select `- No Group Assigned -`.
 - g. Click **Apply Changes**.
2. When you have deleted all pages in the group, click the **Page Groups** tab.
3. Select the group to be deleted.
4. Click **Delete**.

A confirmation page appears.
5. Confirm your request.

Locking and Unlocking a Page

Prevent conflicts during application development by locking pages in your application. By locking a page, you prevent other developers from editing it.

- [Determining If a Page Is Locked](#)
- [Locking a Page](#)
- [Unlocking a Page](#)

Determining If a Page Is Locked

A lock icon indicates whether a page is currently locked. If a page is unlocked, the icon appears as an open padlock. If the page is locked, the icon appears as a locked padlock. A lock icon appears on the Application home page and on the page.

To view the lock icon on the Application home page, click the **View Report** icon. A report appears. The lock icon appears on the right side of the page.

| Page ↑⇅ | Name | Updated | Updated By | Page Type | Group | User Interface | Lock | Run |
|---------|-----------------------|----------------|------------|--------------------|------------|----------------|------|-----|
| 0 | Global Page - Desktop | 11 seconds ago | admin | Global Page | Unassigned | Desktop | | |
| 10 | Home | 10 seconds ago | admin | Home | Unassigned | Desktop | | |
| 20 | Employees | Now | admin | Interactive Report | Unassigned | Desktop | | |
| 21 | Oehr Employee | 10 seconds ago | admin | DML Form | Unassigned | Desktop | | |

Locking a Page

Developers can lock a page on the Application home page, Page Locks page, and in Page Designer.

- [Locking a Page from the Application Home Page](#)
- [Locking a Page from the Page Locks Page](#)
- [Locking a Page in Page Designer](#)

Locking a Page from the Application Home Page

To lock a page from the Application Home Page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the search bar, click the **View Report or View Detail** icon.
4. Click the **Lock** icon.
5. Enter a comment in the Comment field.
6. Click **Lock Page(s)**.

Locking a Page from the Page Locks Page

To lock a page from the Page Locks Page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Utilities** icon.
4. On the Page Specific Utilities list, click **Cross Page Utilities**.
5. Click **Page Locks**.
6. Select the appropriate pages and click **Lock Checked**.
7. Enter a comment in the Comment field.
8. Click **Lock Page(s)**.

Locking a Page in Page Designer

The Page Lock button on the Page Designer toolbar indicates whether a page is locked. If the page is unlocked, the Page Lock button displays as an unlocked padlock.

To lock a page in Page Designer:

1. Navigate to the appropriate page in Page Designer.
2. On the Page Designer toolbar, click the **Page Unlocked** button.
3. Enter a comment in the Comment field.
4. Click **Lock**.

When a page is locked, the Page Lock button displays as a locked padlock.



See Also:

["Viewing a Page in Page Designer"](#)

Unlocking a Page

Only the developer who locks a page can unlock it. However, a developer with administrative privileges can unlock pages locked by other developers.

Tip:

To view information about an existing page lock, navigate to the Lock Details page and expand the Lock History region.

- [Unlocking Pages from Utilities](#)
- [Unlocking Pages from the Application Home Page](#)
- [Unlocking a Page in Page Designer](#)
- [Unlocking Pages As a Workspace Administrator](#)

Unlocking Pages from Utilities

To unlock a page from the Page Locks page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Utilities** icon.
4. On the Page Specific Utilities list, click **Cross Page Utilities**.
5. Click **Page Locks**.
6. Select the appropriate pages.

7. Click **Unlock Checked**.

Unlocking Pages from the Application Home Page

To unlock a page from the Application Home Page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the search bar, click the **View Report** icon.
4. Click the **Lock** icon.

The Lock Details page appears.

5. Click **Unlock**.

Unlocking a Page in Page Designer

The Page Lock button on the Page Designer toolbar indicates whether a page is locked. If the page is locked, the Page Lock button displays as a locked padlock. Passing your cursor over a locked Page Lock button displays a tooltip which indicated who locked it.

To unlock a page in Page Designer:

1. Navigate to the appropriate page in Page Designer.

If a page is locked, the Page Lock button displays as a locked padlock.

2. On the Page Designer toolbar, click the **Page locked by you** button.

If the page has been locked by another user the Lock button appears in red. You cannot unlock a page locked by another user.

3. Click **Unlock**.

When a page is unlocked, the Page Lock button displays as an unlocked padlock.

["Viewing a Page in Page Designer"](#)

Unlocking Pages As a Workspace Administrator

To unlock pages as a Workspace Administrator:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Utilities** icon.
4. On the Page Specific Utilities list, click **Cross Page Utilities**.
5. Click **Page Locks**.
6. Click **Lock Administration**.
7. Select the appropriate pages.
8. Click **Unlock Page(s)**.

8

Editing Pages in Page Designer

Developers view and edit application pages in Page Designer. Page Designer is a full featured Integrated Development Environment (IDE) that includes a toolbar and multiple panes.

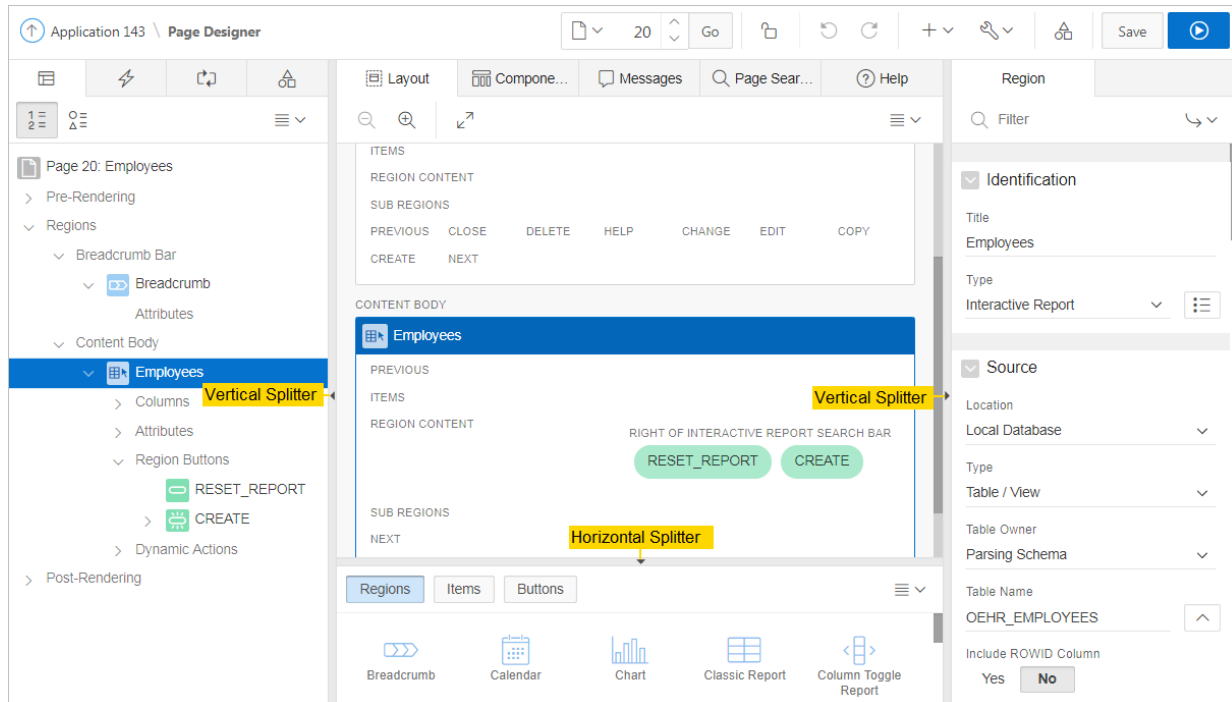
- [About Page Designer](#)
Use Page Designer to maintain and enhance pages within an Oracle Application Express application. Page Designer is a full featured Integrated Development Environment (IDE) that includes a toolbar and multiple panes.
- [Understanding Page Designer UI Elements](#)
The Page Designer window features three panes: a left pane, a central pane, and a right pane. You select a component in the left or central pane and then edit the component's attributes in the Property Editor located in the right pane.
- [Using Page Designer](#)
Use Page Designer to view and edit database application pages.
- [Managing Page Attributes](#)
Edit page attributes to control basic information and functionality for a given page, including the page name or title, header and footer text, navigation menus, inclusion of page-level JavaScript or CSS files, error handling, and the selected authorization scheme.
- [Running a Page from Page Designer](#)
The Application Express engine dynamically renders and processes pages based on data stored in database tables. To view a rendered version of an application, you run or submit it to the Application Express engine.
- [Runtime Developer Toolbar](#)
Use the Developer toolbar to quickly edit the current application or currently running page, or view session state.
- [Copying a Database Application Page](#)
You can copy a page from the current application or from another application. During the copy process, you can also copy shared components or change mappings to shared components in the target application.
- [Deleting Pages](#)
Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries.

See Also:

- ["Adding Developer Comments"](#)
- ["Creating Database Applications"](#)
- ["Controlling Page Layout"](#)
- ["Adding Navigation"](#)

About Page Designer

Use Page Designer to maintain and enhance pages within an Oracle Application Express application. Page Designer is a full featured Integrated Development Environment (IDE) that includes a toolbar and multiple panes.



The Page Designer window is divided into three main panes:

- **Left Pane** - Includes four tabs that display as a tree: Rendering, Dynamic Actions, Processing, and Shared Components.
- **Central Pane** - Includes five tabs: Layout, Component View, Messages, Page Search, and Help.
- **Right Pane** - Displays the Property Editor. Use the Property Editor to update attributes for the selected component. When you select multiple components, the Property Editor only displays common attributes. Updating a common attribute updates that attribute for all of the selected components.

You can adjust the size of each pane by selecting and dragging the horizontal and vertical splitters. To expand or collapse each pane, click the small triangle labeled Collapse in the center of each splitter.

Tip:

You can view an online summary that describes the Page Designer window. View a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.

 **See Also:**

- ["Understanding Page Designer UI Elements"](#)
- ["Viewing Help in Page Designer"](#)
- ["Editing Pages in Page Designer"](#)

Understanding Page Designer UI Elements

The Page Designer window features three panes: a left pane, a central pane, and a right pane. You select a component in the left or central pane and then edit the component's attributes in the Property Editor located in the right pane.

The Page Designer window features three panes: a Left pane, a Central pane, and a Right pane.

The **left** pane includes four tabs: Rendering, Dynamic Actions, Processing, and Page Shared Components. The **central** pane contains four tabs: **Layout**, Component View, Messages, Page Search, and Help and a Gallery pane at the bottom. The **right** pane contains the Property Editor.

 **Tip:**

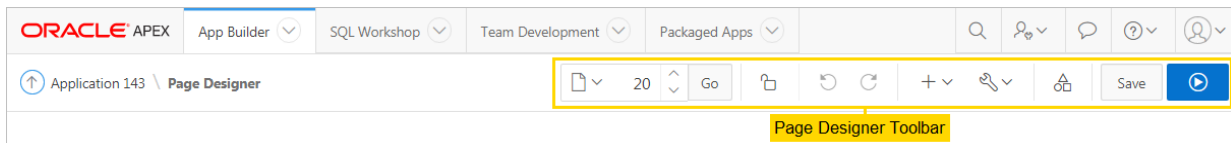
You can view an online summary of this information. View a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.


- [Page Designer Toolbar](#)
The Page Designer toolbar displays at the top of the page and contains both buttons and menu options. Passing your cursor over an active button and menu displays a tooltip.
- [Left Pane of Page Designer](#)
The left pane in Page Designer includes four tabs: Rendering, Dynamic Actions, Processing, and Page Shared Components. Each tab displays a list of the corresponding component types and components created on the current page.
- [Central Pane of Page Designer](#)
The central pane in Page Designer contains five tabs: Layout, Gallery, Component View, Page Search, and Help. The topics that follow provide detailed discussion of how to use these tabs when editing a page.
- [Right Pane of Page Designer \(Property Editor\)](#)
The right pane in Page Designer contains the Property Editor. Use the Property Editor to edit the attributes of the currently selected component.
- [Code Editor](#)
For attributes requiring large amounts of code (for example, a region Source), you can enter text directly in to the field or access a full featured Code Editor.

 **See Also:**
"Using Page Designer"

Page Designer Toolbar

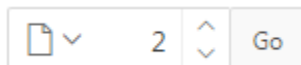
The Page Designer toolbar displays at the top of the page and contains both buttons and menu options. Passing your cursor over an active button and menu displays a tooltip.




 **Tip:**
You can view an online summary of how to use Page Designer. View a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.

The Page Designer toolbar features the following buttons and menu options.

Page Selector



The **Page Selector** displays the current page. Click the down arrow (labeled **Page Finder**) to search for pages. Alternatively, enter a page number in the field and click **Go**. To navigate to the previous or next page, click **Navigate to Next Page** (up arrow) and **Navigate to Previous Page** (down arrow).

 **See Also:**
"Searching for Pages."

Page Unlocked and Page Locked



Indicates the lock status of a page. **Page Unlocked** indicates the current page is unlocked and editable. **Page Locked** indicates the page is unavailable for editing. The appearance of the padlock icon changes depending upon the lock status:

- Clear unlocked padlock — Indicates the page is unlocked.
- Green locked padlock — Indicates the page is locked by you.
- Red locked padlock — Indicates the page is locked by another user. To view the lock owner, or update the lock comment, click the red packlock.



See Also:

"[Locking and Unlocking a Page.](#)"

Undo and Redo



Applies to actions that result in a change to the page data. **Undo** reverts the previous update you made within Page Designer. **Redo** reapplies the last update that was undone using **Undo**.

Create



Features a graphical plus sign (+). Create menu options include:

- **Page.** Access to the Create Page Wizard. See "[About Creating New Pages.](#)"
- **Page as Copy.** Access the Copy Page Wizard. See "[Copying a Database Application Page.](#)"
- **Page Component.** Provides a summary of how to create page components in Page Designer.
- **Form Region.** Access to the Create Form Region Wizard. See "[Creating a Form Region.](#)"
- **Breadcrumb Region.** Access to the Create Breadcrumb Wizard. See "[Creating Breadcrumbs](#)" and "[Creating a Breadcrumb Region Using the Create Breadcrumb Wizard.](#)"
- **Shared Component.** Access to the Create Application Component Wizard. Shared components are common elements that can display or be applied on any page within an application. See "[Managing Shared Components.](#)"
- **Page Group.** Links to the Page Group page. Use page groups to organize pages. See "[Creating Page Groups.](#)"
- **Developer Comment.** Access the Developer Comments dialog. Developers can add comments to an application, a page, or a group of pages. See "[Adding Developer Comments.](#)"
- **Team Development.** Displays the submenus Feature, To Do, and Bug. See "[Tracking Features,](#)" "[Tracking To Dos,](#)" and "[Managing Bugs.](#)"

Utilities



Features a graphic of a wrench. Utilities menu options include:

- **Delete Page.** Delete the current page. See "[Deleting Pages.](#)"
- **Advisor.** Access Oracle Application Express Advisor (Advisor). Use Advisor to check the integrity and quality of your Oracle Application Express application. See "[Running Advisor to Check Application Integrity.](#)"
- **Caching.** Links to the Caching page. Enabling caching is an effective way to improve the performance. See "[Utilizing Region Caching in Page Designer](#)"
- **Attribute Dictionary.** Access the Attribute Dictionary. See "[Using the Attribute Dictionary.](#)"
- **History.** Displays a report of changes to the current page.
- **Export.** Export the current page. See "[Exporting an Application and Application Components.](#)"
- **Cross Page Utilities.** Access Cross Page Utilities. See "[Accessing Cross Page Utilities.](#)"
- **Application Utilities.** Access Application Utilities. See "[Using Application Utilities](#)"
- **Page Groups.** Links to the Page Group page. Use page groups to organize pages. See "[Creating Page Groups.](#)"
- **Upgrade Application.** Upgrade an existing application. See "[Running Upgrade Application.](#)"
- **Show Tooltips.** Disable and enable tooltips within the UI.
- **Layout.** Controls the number of panes that display in Page Designer. See "[Switching Between Three Pane and Two Pane Mode.](#)"

Team Development



Features a graphical outline of two people. Options include:

- **Features.** Links to Create Feature. See "[Tracking Features.](#)"
- **To Do -** Links to Create To Do. See "[Tracking To Dos.](#)"
- **Bugs -** Links Create Bug. See "[Managing Bugs.](#)"

- **Feedback Entries -** Links to the Feedback report. See "[Tracking Feedback Team in Development.](#)"

 **Tip:**

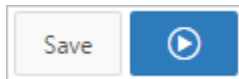
This button only displays if Team Development must be enabled for the workspace or there is a feature, To Do, bug, or feedback entry linked to the current page. See "Managing Team Development" in *Oracle Application Express Administration Guide*.

Shared Components



Links to the Shared Components page. Shared components can display or be applied on any page within an application. See "[Managing Shared Components](#)."

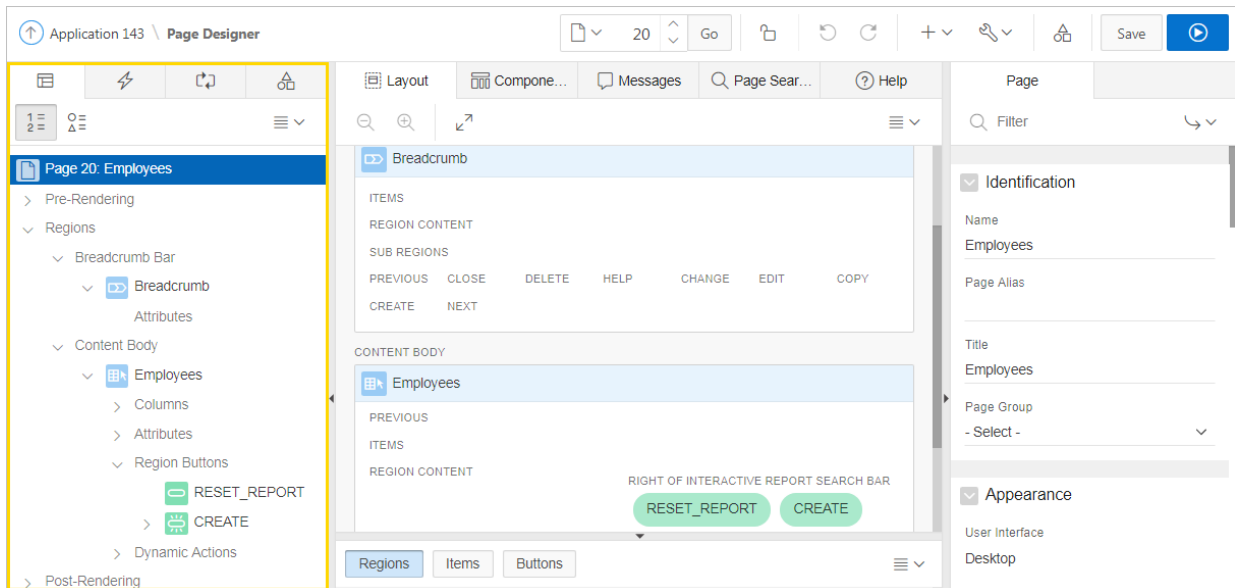
Save and Save and Run Page



Click **Save** to save the current page. Click **Save and Run Page** to save and then run the current page. See "[Running a Page from Page Designer](#)."

Left Pane of Page Designer

The left pane in Page Designer includes four tabs: Rendering, Dynamic Actions, Processing, and Page Shared Components. Each tab displays a list of the corresponding component types and components created on the current page.



Key features of the Left pane include:

- **Context menus.** Right-click a component or control to display a context menu.

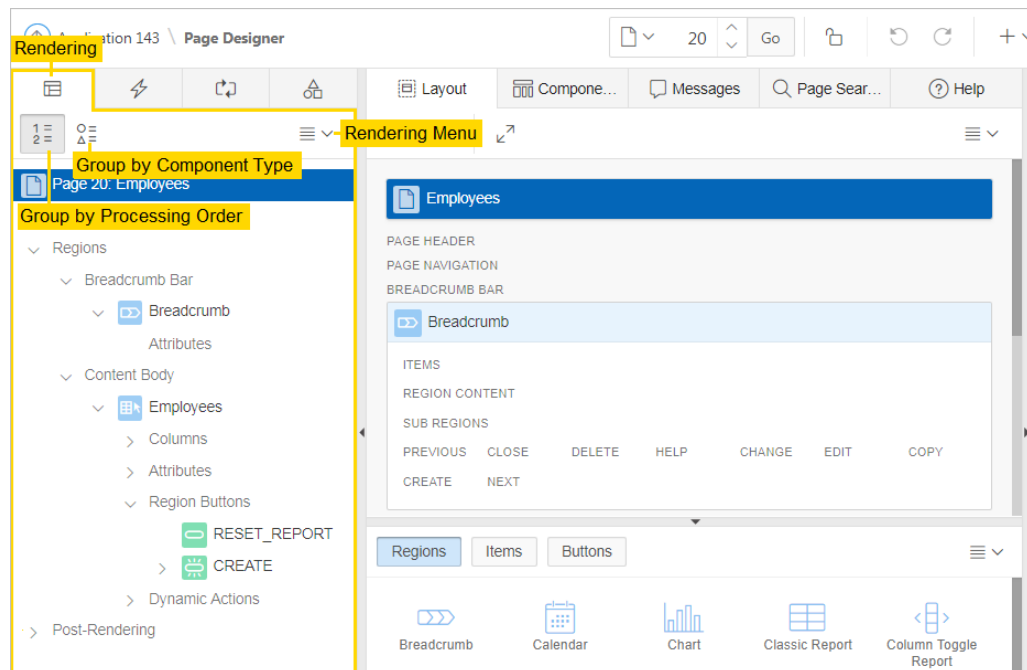
- **Quick Access to the Property Editor.** Select a component or multiple components to display the corresponding attributes in the Property Editor in the right pane.
- **Drag and Drop.** Drag components up and down within the tree to change the position or sequence of the selected component.
- **Tooltips.** Position the mouse over a component or control to view a tooltip of basic information, such as the region type, item type, and so on.

Rendering Tab

The Rendering tab displays regions, page items, page button, page components, and application logic. The Rendering tab groups and orders components based on how Oracle Application Express processes them.

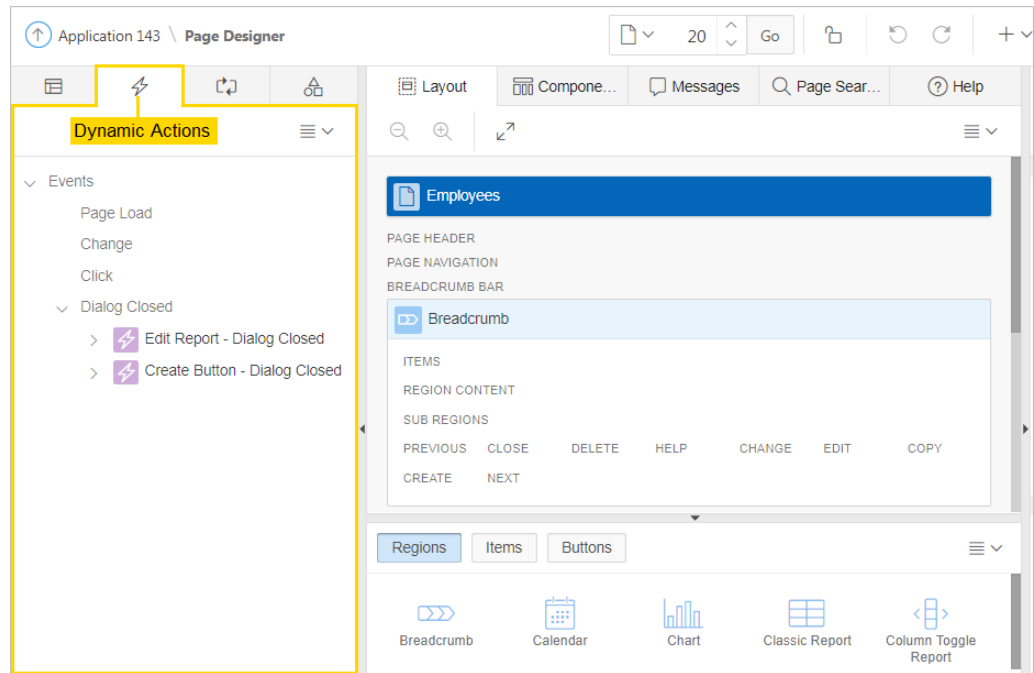
You can alter the display by clicking the following icons:

- **Group by Processing Order**
- **Group by Component Type**
- Rendering Menu: **Expand All Below** or **Collapse All Below**



Dynamic Actions Tab

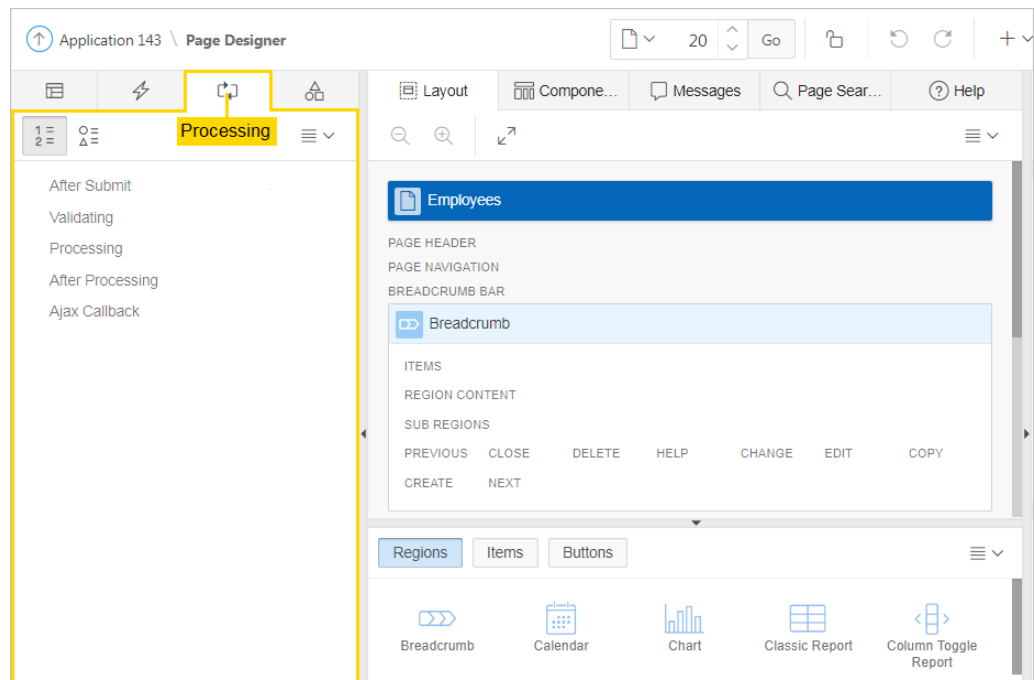
The Dynamic Actions tab displays dynamic actions defined on the current page. By creating a dynamic action, you can define complex client-side behavior declaratively without the need for JavaScript.



 **See Also:**
"Managing Dynamic Actions"

Processing Tab

The Processing tab displays application logic defined on the page and groups and orders components based on how Oracle Application Express processes them.



You can alter the display by clicking the following icons:

- **Group by Processing Order**
- **Group by Component Type**
- Processing Menu: **Create Computation**, **Expand All Below**, or **Collapse All Below**

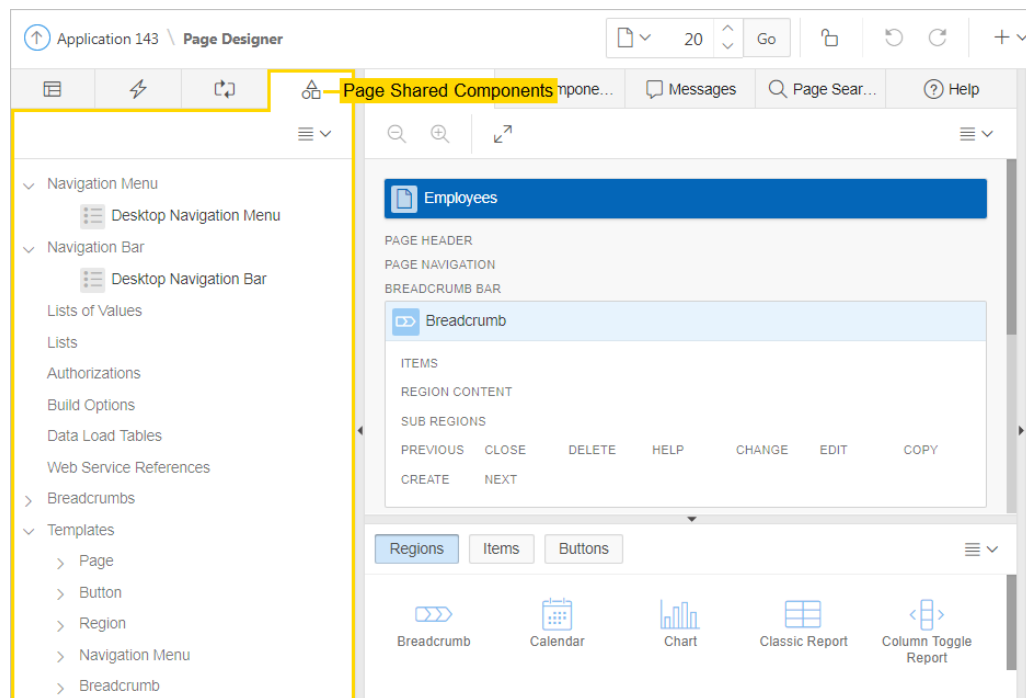


See Also:

["Managing Computations, Validations, and Processes"](#)

Page Shared Components Tab

The Page Shared Components tab displays shared components associated with this page.



See Also:

["Managing Shared Components"](#)

Central Pane of Page Designer

The central pane in Page Designer contains five tabs: Layout, Gallery, Component View, Page Search, and Help. The topics that follow provide detailed discussion of how to use these tabs when editing a page.

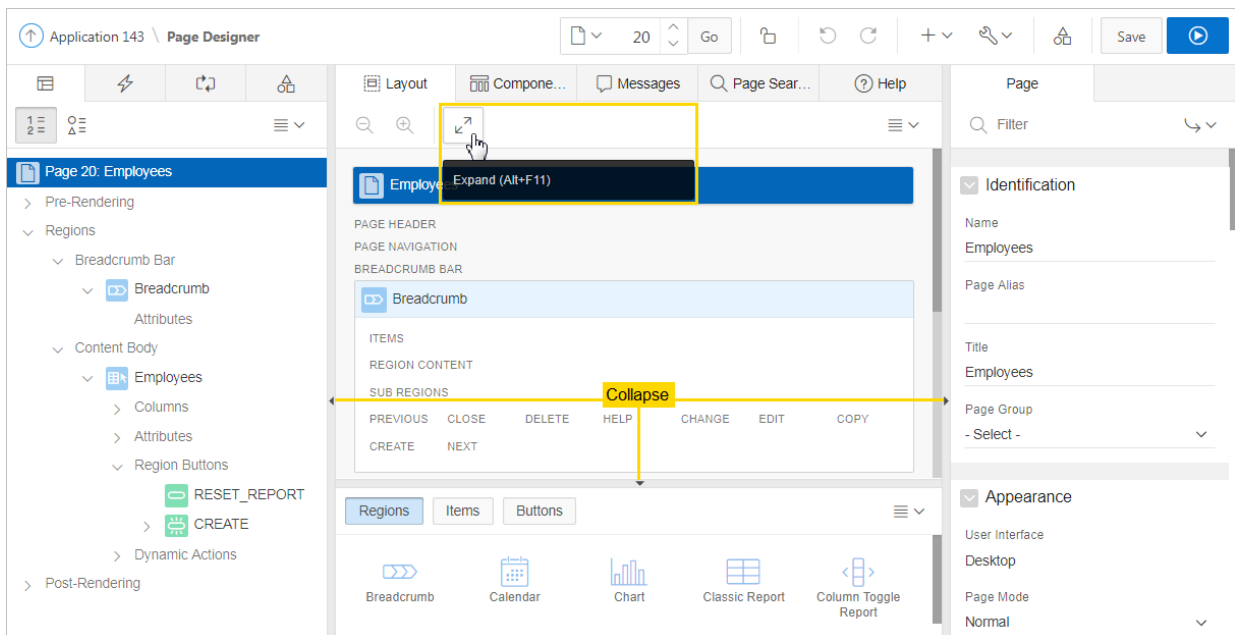
- [Layout Tab](#)

- Gallery
- Component View Tab
- Messages Tab
- Page Search Tab
- Help Tab

Layout Tab

The Layout tab displays in the central pane of Page Designer and is a visual representation of how the components are positioned on the page. The Layout tab features context menus. By selecting a component and right-clicking, you can delete, move, or copy the component to other regions, or new regions on the page. You can also move existing regions, items, and buttons relative to other components by simply clicking on the component and dragging it to the new location. For example, you can place items next to one another by dragging the second item to the end of the first item and dropping it in the dark yellow box that appears when you hover in the desired location. You can quickly add new components to an existing page by dragging the component from the Gallery up to the desired position within the Layout tab.

You can adjust the size of each pane by selecting and dragging the horizontal and vertical splitters. To expand or collapse each pane, click the small triangle labeled Collapse in the center of each splitter. Alternatively, you can click the **Expand** button to make the Layout tab larger and then **Restore** to return it to the previous size.



Tip:

Hidden items do not display in the Layout tab, but do display in the Rendering tree.

Key features of the Layout tab include:

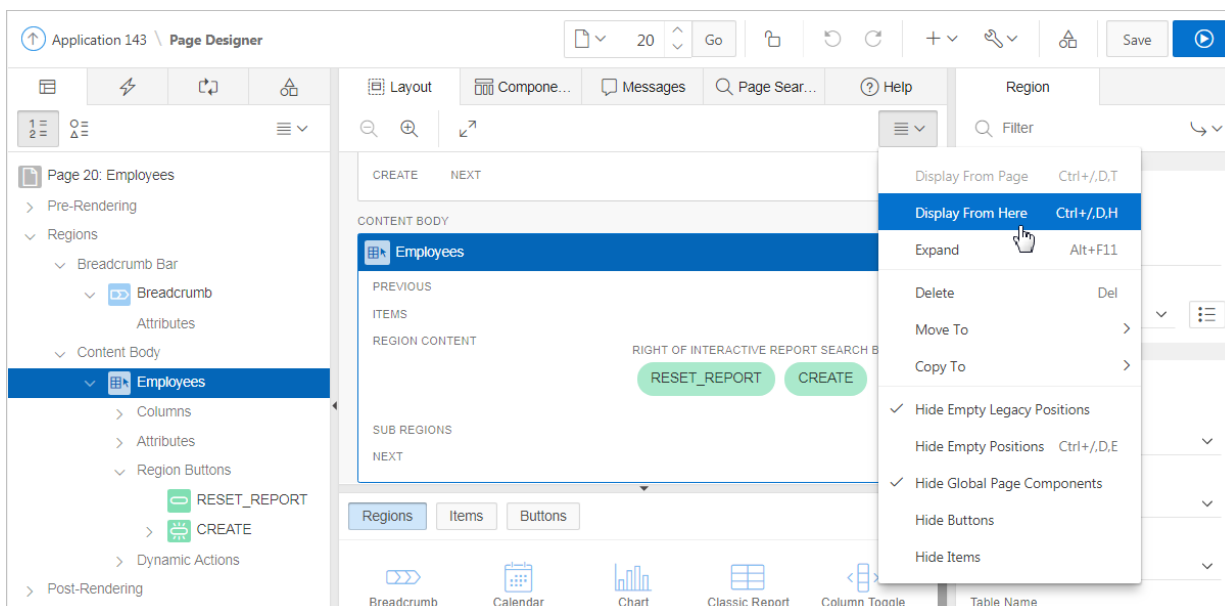
- **Context menus.** Right-click a component or control to display a context menu.
- **Quick Access to the Property Editor.** Select a component or multiple components to display the corresponding attributes in the Property Editor in the right pane.
- **Drag and Drop .** Move, copy, and reorder regions, items, and buttons by dragging and dropping. You can also add new regions, items, and button by dragging them from the Gallery and dropping them to the desired position on the page.
- **Tooltips.** Position the mouse over a component or control to view a tooltip of basic information, such as the region type, item type, and so on.

 **Tip:**

When dragging components to a new position, such as dragging an item onto the same line as an exiting item, you need to drag the component into the desired position and wait until the grid changes and the new drop position displays. See "[Adding a Region, Item, or Button by Dragging and Dropping](#)".

Layout Menu

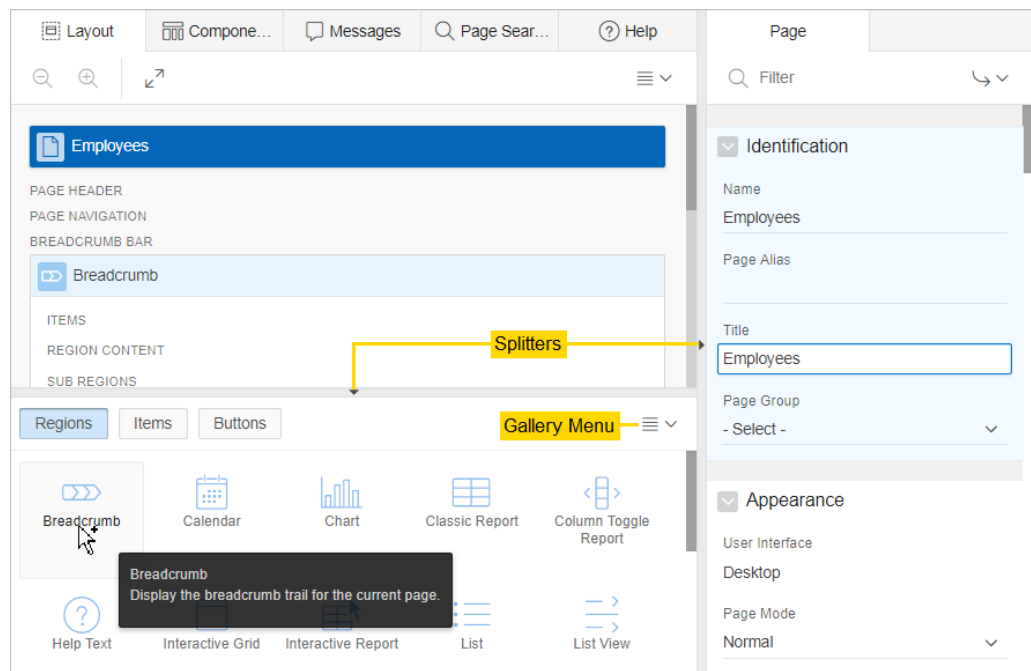
The Layout menu displays on the right side of the pane. Use this menu to customize the type and amount of information that displays and to delete, move, and copy selected components. To have a specific region fill the the Layout tab, select the region and select **Display from Here** from the Layout menu. The selected region fills the Layout tab. To restore the view, select **Display from Page**.



See Also:["Adding a Region, Item, or Button by Dragging and Dropping"](#)

Gallery

The Gallery displays beneath the Layout tab in the central pane of Page Designer. The Gallery contains three tabs: Regions, Items, and Buttons. When you pass the cursor over a control or component, a tooltip displays that describes it. You can adjust the size of the pane by dragging the horizontal and vertical splitters. To collapse or restore the Gallery pane, click the small triangle in the center of each splitter.



You can add new controls and components to a page by selecting the control or component from the Gallery and dragging and dropping them into the Layout tab. Alternatively, you can select a control or component in the Gallery and right-click to view a context menu.

See Also:["Adding a Region, Item, or Button by Dragging and Dropping"](#) and ["About Adding Components Using Gallery Context Menus"](#)

Gallery Menu

The Gallery menu displays on the right side of the Gallery pane. By default, the Gallery only displays supported controls and components. To view unsupported controls and components (such as legacy application components), enable **Show Unsupported**

Components on the Gallery menu. To view legacy controls and components, enable **Show Legacy Components** on the Gallery menu.



Note:

"Managing Application Legacy Components"

Component View Tab



Tip:

The Component View tab was added in Oracle Application Express release 5.1 to assist developers with the transition from Legacy Component View to Page Designer. Legacy Component View was desupported in release 18.1.

The Component View tab displays in the central pane of Page Designer and presents user interface elements and application logic by component type. The Component View tab consists of following sections:

- **Page Rendering.** Page rendering is the process of generating a page from the database. Use the Page Rendering section to modify controls that impact the rendering of a page, including page attributes, regions, buttons, items, page rendering computations, and page processes.
- **Page Processing.** Page processing is the process of submitting a page. A page is typically submitted when a user clicks a button. Use the Page Processing section to specify application logic such as computations, validations, processes, and branches. In general, the Application Express engine runs this logic in the order it appears on the page.

You can adjust the size of the Component View tab using the window splitters. Alternatively, you can click the **Expand** button in the upper left corner to maximize the window and then click **Restore** to return it to the previous size.

To edit a component, click the component name and the Property Editor highlights the corresponding attribute in the right pane. Unlike the Layout tab, the Component View tab does not offer a visual representation of a page or support the dragging and dropping of page components. To preview the position of components on a page, click **Layout** or run the page.

To add a new control, component, or process, click the **Create** icon to the right of the section name.

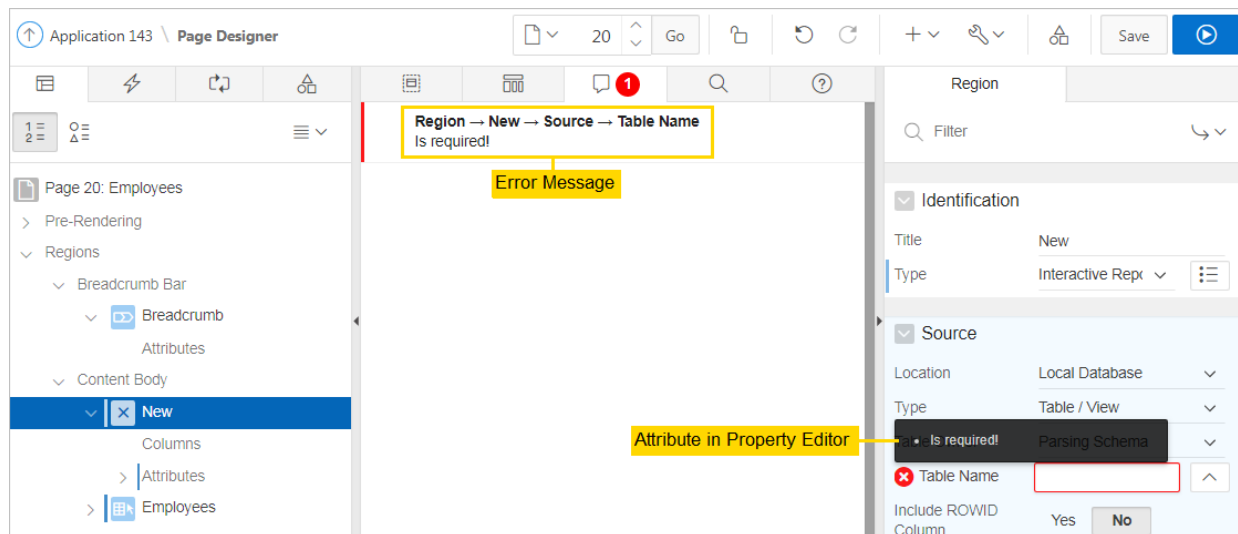


See Also:

"Adding a Region, Item, or Button from the Component View Tab"

Messages Tab

The Messages tab displays in the central pane of Page Designer. When you create components or edit attributes in Page Designer, the Messages tab displays errors and warnings you need to address. The Messages tab displays a red or yellow badge indicating the number of messages you need to address. Clicking on a message changes the focus within Property Editor to the corresponding attribute associated with the error or warning. To view Help, click the Help tab and then select an attribute within the Property Editor located in the right pane.



The Message tab displays two types of messages:

- **Errors** - Error messages display in red. Selecting an error message displays the associated attribute in red in the Property Editor. You must address errors before a page can be saved.
- **Warnings** - Warning messages display in yellow. Selecting a warning message displays the associated attribute in yellow in the Property Editor. You can save a page without addressing warning messages.

Tip:

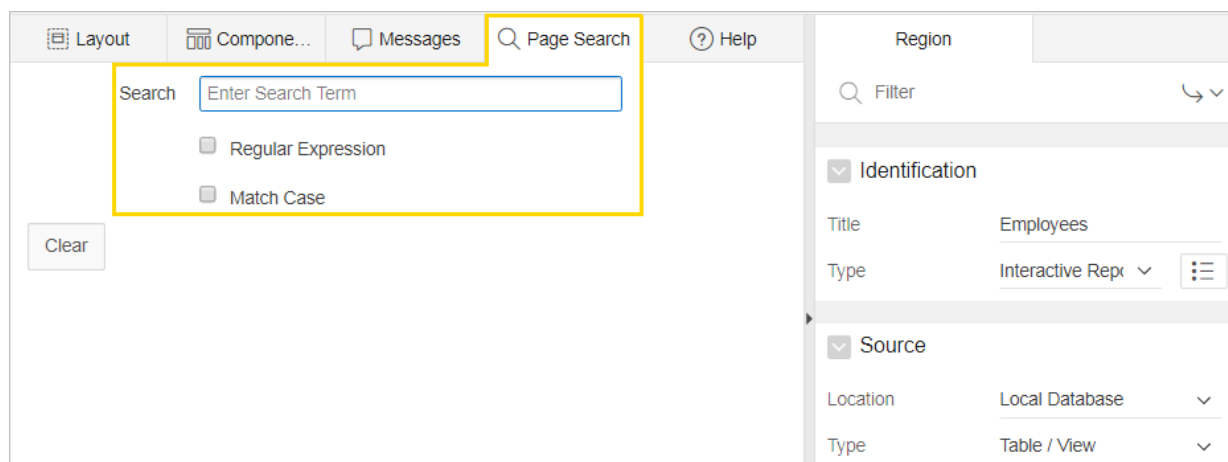
Components that have errors or warnings display in red with a X icon or yellow with a triangle icon in both the left pane and the Layout tab.

See Also:

"Adding a Region, Item, or Button by Dragging and Dropping" and "About Adding Components Using Gallery Context Menus"

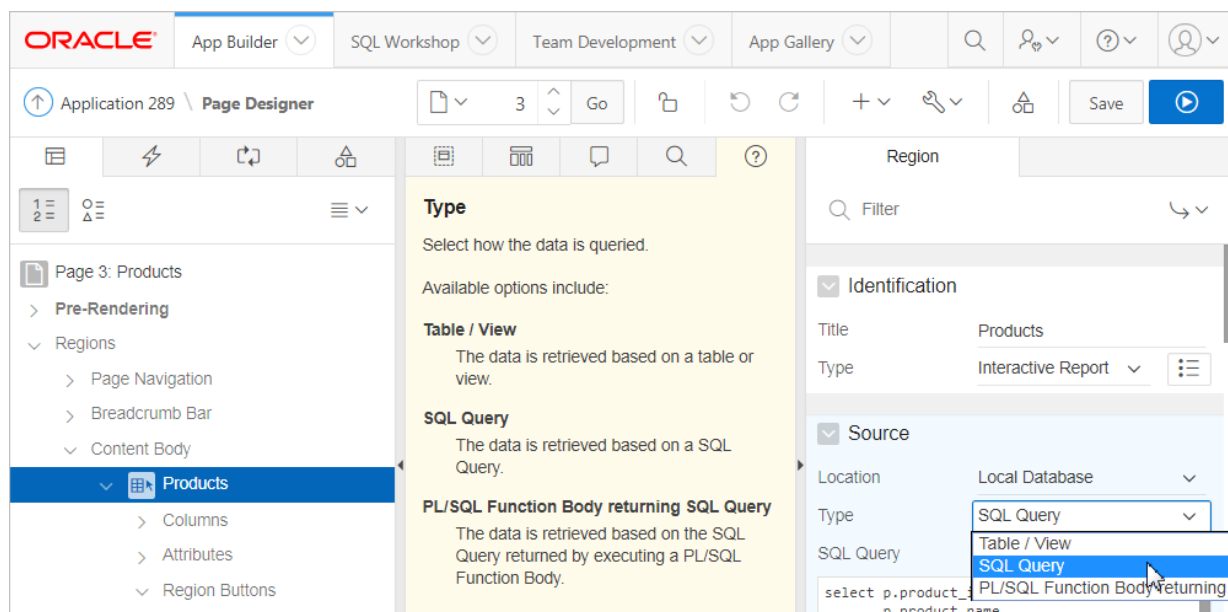
Page Search Tab

The Page Search tab displays to the right of Messages in the central pane. Use Page Search to search all page metadata including regions, items, buttons, dynamic actions, columns, and so on. To search a page, enter a search terms in the field provided. To match the case, select **Match Case**. To search for a regular expression, select **Regular Expression**.



Help Tab

Page Designer includes Help for every Property Editor attribute. To view Help, select the attribute and click the **Help** tab. Once you activate the Help pane, the content that displays changes every time you select another attribute. In this example, the **Source, Type** attribute is selected and the Help tab describes what the attribute is for and the available options.



 **See Also:**

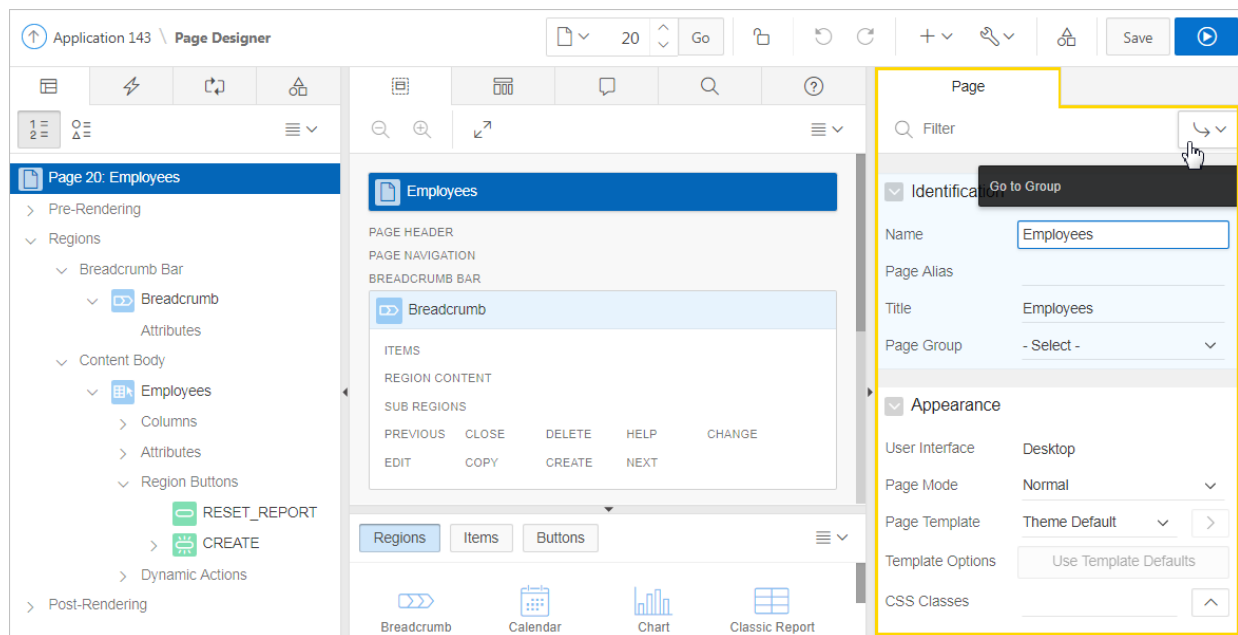
"Viewing Help in Page Designer"

Right Pane of Page Designer (Property Editor)

The right pane in Page Designer contains the Property Editor. Use the Property Editor to edit the attributes of the currently selected component.

Property Editor

Property Editor displays all attributes for the currently selected component. When you select multiple components, the Property Editor only displays common attributes. Updating a common attribute updates that attribute for all of the selected components.

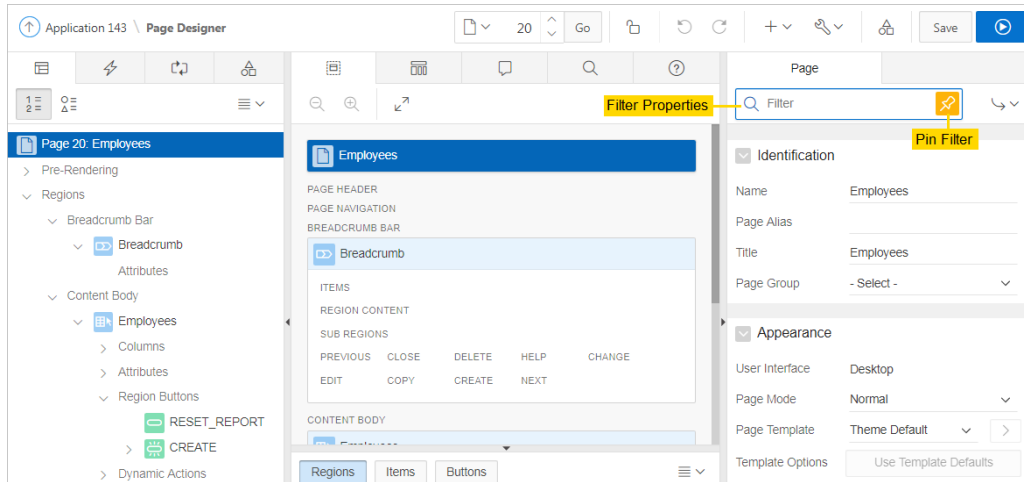



Property Editor organizes attributes into functional groups. To quickly access a group, click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

Searching for Attributes

To search for a group or an attribute, enter a keyword in the Filter Properties field. The groups and attributes containing the keyword appear. To return to the default display, delete the keyword in the Filter Properties field. To search for keywords across all items, regions, and processes, click the **Pin Filter** button on the right side of the search field. When activated, you can search terms you enter remain persistent as you click around Page Designer. Click Pin Filter again to disable

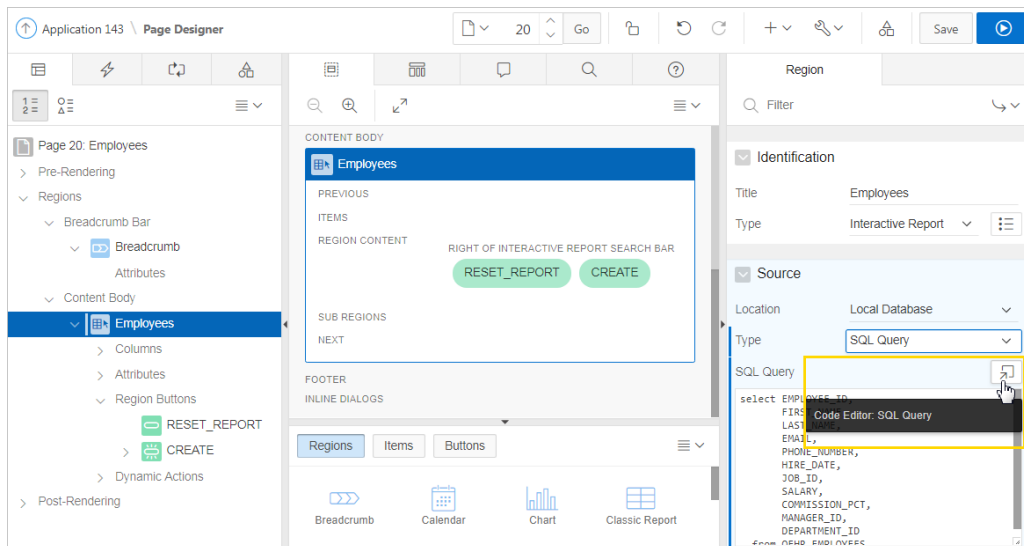
Click the **Pin Filter** button on the right side of the search field to search across all items, regions, and processes. You can “pin” keywords to have them persist as you click around Page Designer. To “unpin” the search, click the **Pin Filter** button again.



 **See Also:**
["Using Page Designer"](#) and ["Managing Page Attributes"](#)

Code Editor

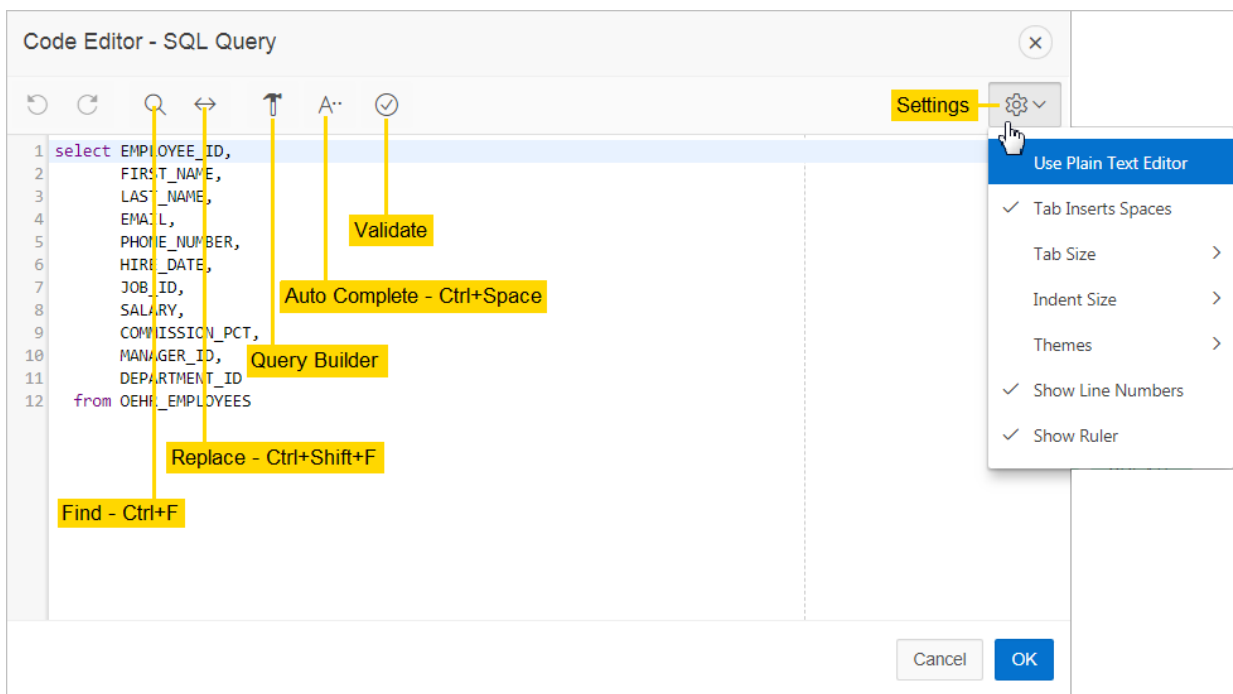
For attributes requiring large amounts of code (for example, a region Source), you can enter text directly in to the field or access a full featured Code Editor.



 **See Also:**

"Accessing Property Editor"

You access Code Editor by clicking the **Code Editor** button which resembles an arrow pointing at a square. The Code Editor displays as a modal dialog and provides an enhanced code editor to edit PL/SQL, SQL, HTML, CSS, and JavaScript component properties. The code highlighting is determined by the input required for the specific property.



Key features of the Code Editor include:

- Undo - Ctrl+Z
- Redo - Ctrl+Shift+Z
- Find - Ctrl+F
- Replace - Ctrl+Shift+F
- Query Builder
- Auto Complete - Ctrl+Space
- Validate
- Syntax highlighting
- Block indent and unindent
- Resize dialog

The **Settings** menu located on the right side offers additional options including: **Use Plain Text Editor**, **Tab Insert Spaces**, **Tab Size**, **Indent Size**, **Themes**, **Show Line Numbers**, and **Show Ruler**.



Tip:

To switch to a plain text editor, select **Use Plain Text Editor** from the Settings menu.

Using Page Designer

Use Page Designer to view and edit database application pages.



Tip:

To view a list of keyboard shortcuts, click **Alt+Shift+F1**. Alternately, click the **Help** menu at the top of the page and select **Shortcuts**.

- [Viewing a Page in Page Designer](#)
- [Searching for Pages](#)
- [Dragging and Dropping Tabs Between Panes](#)
- [Switching Between Three Pane and Two Pane Mode](#)
- [Accessing Property Editor](#)
- [Adding a Region, Item, or Button by Dragging and Dropping](#)
- [About Adding Components Using Gallery Context Menus](#)
- [Adding a Region, Item, or Button from the Rendering Tab](#)
- [Adding a Region, Item, or Button from the Component View Tab](#)
- [Copying a Region, Item, or Button](#)
- [Creating Pre-Rendering Branches, Computations, and Processes](#)
- [Creating Page Submission Branches, Validations, Computations, and Processes](#)



See Also:

"[Understanding Page Designer UI Elements](#)" and "[Viewing Help in Page Designer](#)"

Viewing a Page in Page Designer

To view a page in Page Designer:

1. Navigate to the Workspace home page.

2. Click the **App Builder** icon.
3. Select an application.
4. Select a page.

The default view, Page Designer appears.

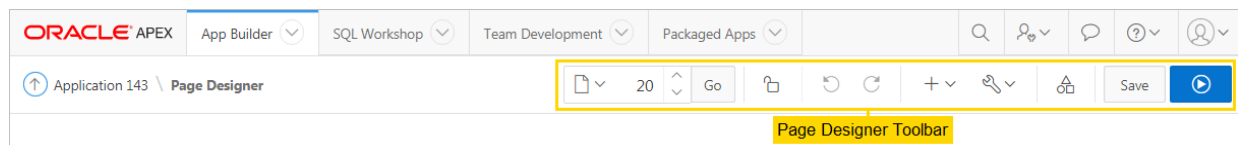
You preview a page by running it. Running an application page displays the rendered version as end users would see it.



See Also:

["Running a Page from Page Designer"](#)

Searching for Pages



The Page Selector is the first control in the Page Designer toolbar and displays the current page number. Use the Page Selector to search for pages within the current application.

To search for pages within an application:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. On the Page Designer toolbar, click the **Page Selector**.

The Page Finder appears.

3. Click the tabs at the top of the window to change the view:
 - **Current User Interface** - Displays pages in the current user interface.
 - **Current Group** - If the current page is associated with a group, this tab displays the pages associated with the current group.
 - **All Pages** - Displays all pages in the current application.
 - **Recently Edited** - Displays recently edited pages.
4. Enter search terms in the Search field and press **ENTER** or click **Go**.

Search for page numbers, page names, user interface, or group names. The search is not case sensitive and supports regular expressions.

**See Also:**

"Running a Page from Page Designer" and "Creating Page Groups"

Dragging and Dropping Tabs Between Panes

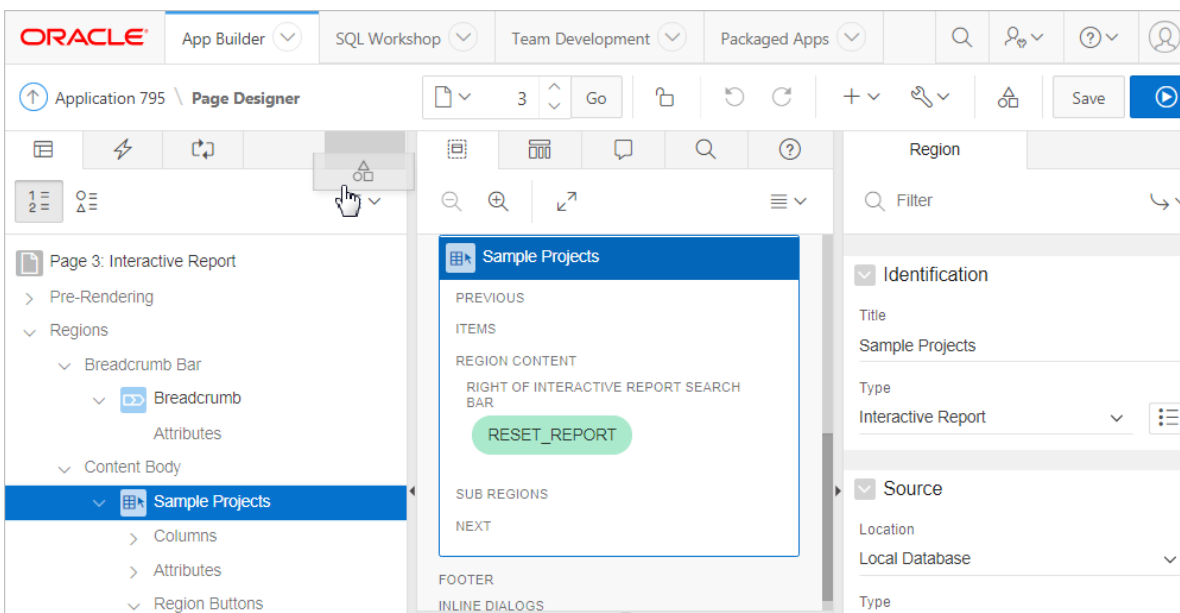
By default, Page Designer displays three panes which contain one or more tabs. You can drag and drop tabs within a pane or between panes by selecting the pane heading (or icon) and then dragging and dropping it to a new location. To return to the default display, click **Utilities** on the Page Designer toolbar and select **Layout** and then **Reset Layout**.

To drag and drop tabs in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. Select a tab heading (or icon) and then drag and drop it to a new location.

In the following example, the **Page Shared Components** tab is selected being moved within the left pane.



3. To return to the default display:
 - a. On the Page Designer toolbar, click **Utilities**.
 - b. Select **Layout** and then **Reset Layout**.

**See Also:**

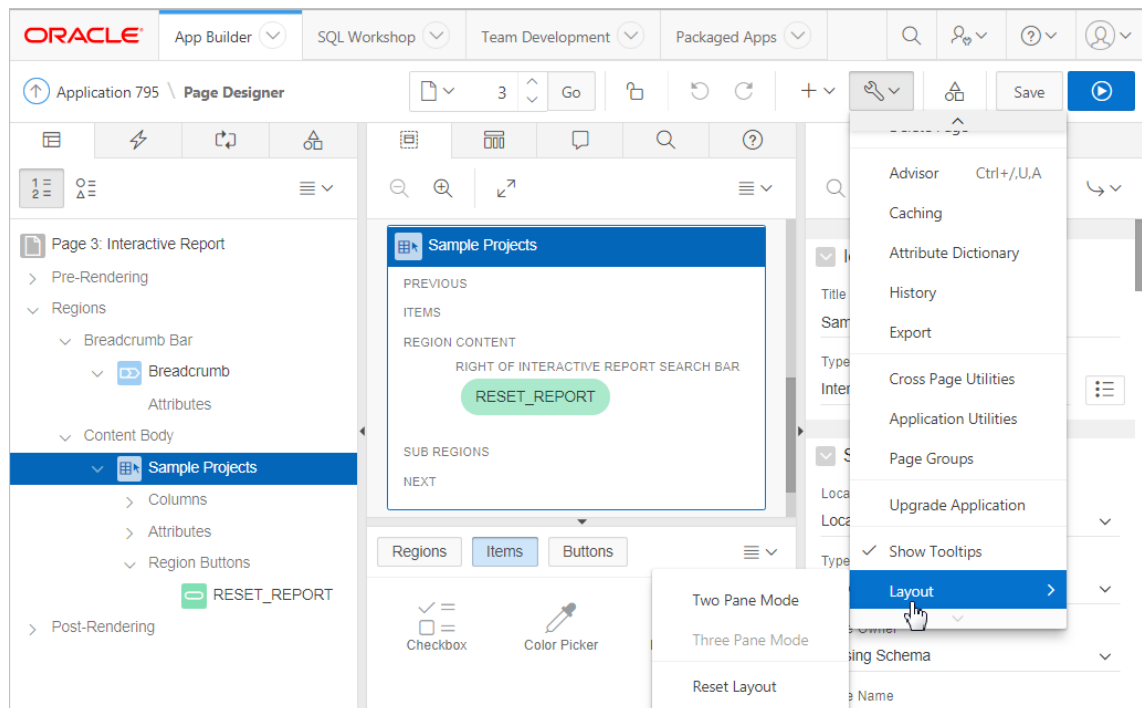
"Understanding Page Designer UI Elements"

Switching Between Three Pane and Two Pane Mode

Use the Layout options on the Utilities menu on the Page Designer toolbar to switch between pane modes. **Three Pane Mode** (the default) displays the left pane, central pane, and the right pane. **Two Pane Mode** displays just the central pane and the right pane. A typical use case for accessing Two Pane Mode, is when you want to view just the Rendering tree and the Property Editor. To return to the default display, click **Utilities** on the Page Designer toolbar and then select **Layout** and then **Reset Layout**.

To hide and show Page Designer panes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. To select a page mode:
 - a. On the Page Designer toolbar, click **Utilities** and then **Layout**.



The Layout menu has three options: **Two Pane Mode**, **Three Pane Mode**, and **Reset Layout**.

- b. From Layout, select one of the following:
 - **Two Pane Mode** - Displays just the central pane and the right pane.
 - **Three Pane Mode** (the default) - Displays the left pane, central pane, and the right pane.
 3. To return to the default display:
 - a. On the Page Designer toolbar, click **Utilities**.
 - b. Select **Layout** and then **Reset Layout**.

Accessing Property Editor

The Property Editor displays in the right pane of Page Designer and displays all attributes for currently selected components.

To access Property Editor:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the left pane or the Layout tab, select components or controls to edit. Passing the cursor over a control or component, displays a tooltip that describes it.

Tip:

Once you select a component, press ALT+6 to navigate directly to the Property Editor. To learn more, see *Oracle Application Express Accessibility Guide*

The Property Editor displays attributes for the selected component or control. As you select different components in either the left pane or the Layout tab, the Property Editor automatically updates to reflect the selected component. When you select multiple components, the Property Editor only displays common attributes. Updating a common attribute updates that attribute for all of the selected components.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

**See Also:**

"Right Pane of Page Designer (Property Editor)"

Adding a Region, Item, or Button by Dragging and Dropping

Each application page can have buttons and fields (called items) which are grouped into containers called regions. You can add regions, items, and buttons to a page by dragging and dropping them from the Gallery.

To add a region, item, or button by dragging and dropping from the Gallery:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. If needed, select the Layout tab in the central pane.
3. In the Gallery at the bottom of the central pane, locate the component or control you want to create.

The Gallery lists all controls or components you can add to a page. Passing the cursor over a control or component displays a tooltip that describes it.

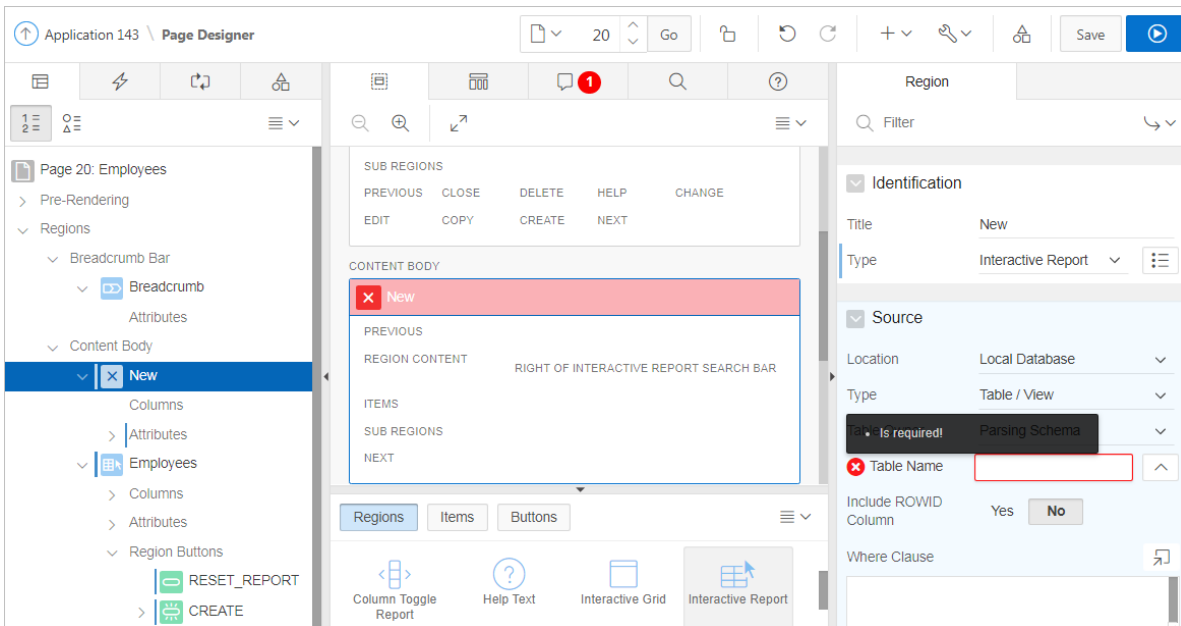
**Tip:**

In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Select **Add To** and then the location where you want to add the component.

4. Click and hold the mouse on the component to be created and drag it the desired location in the Layout tab.

When the mouse is over the appropriate location, the Layout tab displays as a darkened yellow tile. Release the mouse to drop the component. You can only drop components into appropriate drop positions, as determined by the component type.

Based on the type of component you add, Page Designer indicates what actions are required next. The Messages tab displays a red or yellow badge indicating the number of messages you need to address. The following example shows a new Interactive Grid region with an error message indicating that you need to select a table name.



5. Edit the appropriate attributes in the Property Editor.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

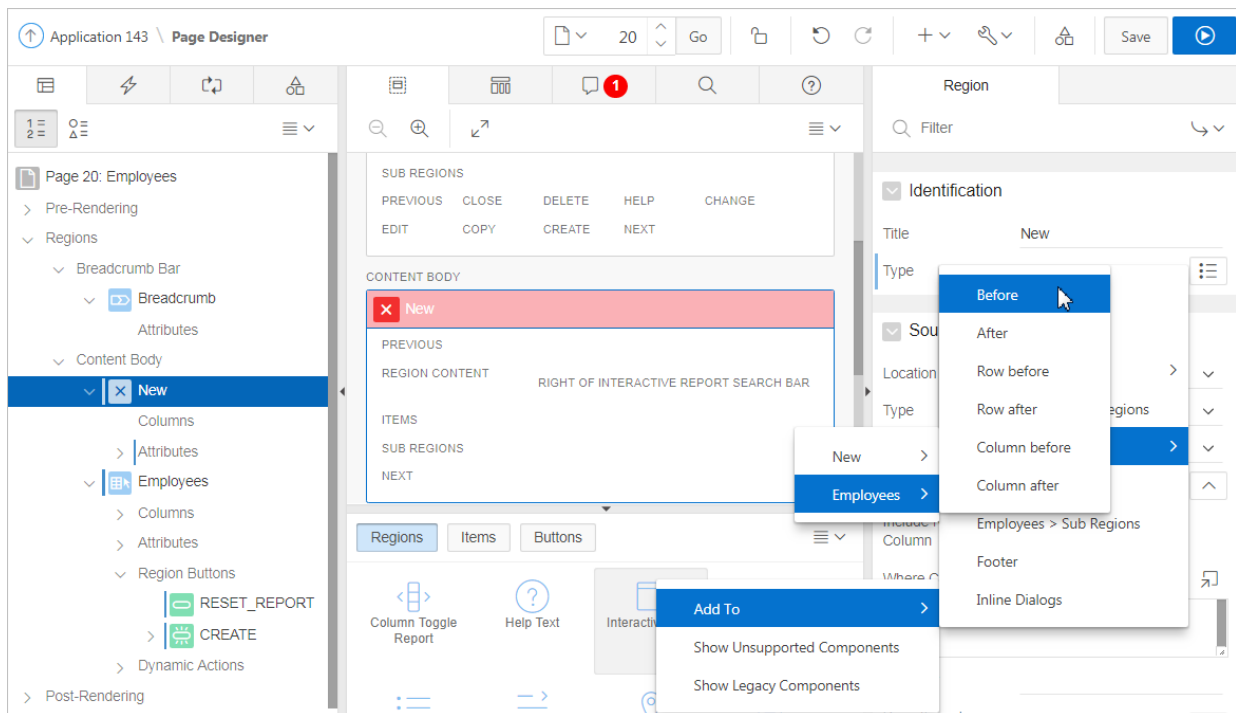
6. Click **Save** or **Save and Run Page**.

See Also:

- ["Messages Tab"](#)
- ["Running a Page from Page Designer"](#)
- ["About Adding Components Using Gallery Context Menus"](#)

About Adding Components Using Gallery Context Menus

In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Each context menu features an Add To option which displays the actual locations where you can add the component. In this example, the developer is adding a new report to before the Content Body region.



See Also:

["Adding a Region, Item, or Button by Dragging and Dropping"](#)

Adding a Region, Item, or Button from the Rendering Tab

To create new components within the Rendering tab:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
2. In the left pane, select the **Rendering** tab.
3. Locate and select the appropriate region, item, or region button.
4. Right-click an existing component or container to display the context menu for that component type.
5. Select a command (for example, Create Region, Create Sub Region, Create Page Item, Create Button and so on). The options available depend upon the node you select.
6. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Click **Save** or **Save and Run Page**.

 **See Also:**

- ["Right Pane of Page Designer \(Property Editor\)"](#)
- ["Managing Regions "](#)
- ["Viewing Page-Level Items"](#)
- ["Creating a Button"](#)

Adding a Region, Item, or Button from the Component View Tab

To assist developers with the transition to Page Designer, Page Designer includes a Component View tab. Similar in appearance to Legacy Component View, the Page Designer Component View tab presents user interface elements and application logic by component type.

To create new components from the Component View tab:

1. View the page in Page Designer. See ["Viewing a Page in Page Designer."](#) Page Designer appears.
2. In the central pane, select the **Component View** tab.
3. Under the appropriate component type, click the **Create** icon.
Page Designer identifies the new component as "New" in the left pane and in the central pane.
4. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save** or **Save and Run Page**.

 **See Also:**

- ["Right Pane of Page Designer \(Property Editor\),"](#)
- ["Managing Buttons"](#)
- ["Managing Regions ,"](#)
- ["Viewing Page-Level Items,"](#)

Copying a Region, Item, or Button

Each application page can have buttons and fields (called items) which are grouped into containers called regions. Developers can copy regions, items, and buttons by using context menus or pressing **CTRL** and dragging and dropping with the mouse.

Copying a Region, Item, or Button Using Context Menus

To copy a region, item, or button using context menus:

1. In the central pane, select the **Layout** tab.
2. Locate and select the appropriate region, item, or region button.
3. Right-click to display the context menu.
4. From the context menu, select **Copy To** and then select the location.
5. Edit the component attributes.

Copying a Region, Item, or Button by Dragging and Dropping

To copy a region, item, or button by dragging and dropping:

1. In the left pane, select the **Rendering** tab.
2. Locate and select the appropriate region, item, or region button.
3. Press **CTRL** and then use the mouse to drag and drop the component to the new location.

A copy of the component appears in the new location.

4. Edit the component attributes.

Copying a Region, Item, or Button to Another Page

To copy a region, item, or button to another page:

1. In the left pane, select the **Rendering** tab.
2. Locate and select the appropriate region, item, or region button.
3. Right-click the component, and select **Copy to other Page**.
The appropriate wizard appears.
4. Follow the on-screen instructions.

Creating Pre-Rendering Branches, Computations, and Processes

To create branches, computations, and processes that execute before rendering the page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the left pane, select the Rendering tab.
3. Expand the Pre-Rendering node at the top of the Rendering tab.
4. Determine the processing point. Right-click **Before Header**, **After Header**, or **Before Regions** to display the context menu for that processing point.
5. Select the appropriate option: **Create Branch**, **Create Computation**, or **Create Process**.
6. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Click **Save** or **Save and Run Page**.

 **Tip:**

If you have changed the default display to group components by component type, then you must to navigate to the Processing tab to maintain pre-rendering components.

 **See Also:**

["Managing Computations, Validations, and Processes"](#)

Creating Page Submission Branches, Validations, Computations, and Processes

To create branches, validations, computations, and processes that execute when the page is submitted:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the left pane, select the Processing tab.
3. Navigate to After Submit, Validating, Processing, or After Processing as appropriate within the processing tree.
4. Right-click the identifier or an existing component to display the context menu for that processing point.
5. Select **Create Branch**, **Create Validation**, **Create Computation**, or **Create Process**.
6. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Click **Save** or **Save and Run Page**.

 **Tip:**

You can use context menus to duplicate, delete, and copy a component to other pages.

 **See Also:**

["Managing Computations, Validations, and Processes"](#)

Managing Page Attributes

Edit page attributes to control basic information and functionality for a given page, including the page name or title, header and footer text, navigation menus, inclusion of page-level JavaScript or CSS files, error handling, and the selected authorization scheme.

- [Editing Page Attributes](#)
- [Incorporating JavaScript Using Page Designer](#)
- [Configuring Rejoin Sessions](#)

- [Configuring Dialog Attributes](#)
- [Configuring Page Attributes to Warn Users of Unsaved Changes](#)



See Also:

["Right Pane of Page Designer \(Property Editor\)"](#)

Editing Page Attributes

To edit page attributes in the Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the page name.
3. Edit the appropriate attributes in the Property Editor.

Edited attributes display a blue marker to the left of the attribute name until the page is saved.



Note:

To learn more about an attribute, see field-level Help.

4. To save your changes, click **Save** or **Save and Run Page**.



See Also:

["Right Pane of Page Designer \(Property Editor\)"](#) and ["Viewing Field-Level Help"](#)

Incorporating JavaScript Using Page Designer

You can include JavaScript on a page by editing the attributes in the JavaScript group. This is a good approach for functions that are very specific to a page and a convenient way to test a function before you include it in a .js file.

To add JavaScript code to the JavaScript attribute:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.

- b. Select an application.
- c. Select a page.

Page Designer appears.

2. In the Rendering tab, select the page name.

The Property Editor displays the Page attributes.

3. In the Property Editor, find the **JavaScript** group.
4. Under JavaScript, edit following attributes:

- a. **File URLs** - Enter JavaScript file URLs for code to be loaded on this page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.
- b. **Function and Global Variable Declaration** - Enter JavaScript code such as functions or global variable declarations referenced on this page. If the same JavaScript is required on multiple pages, consider putting the code into an external JavaScript file to avoid duplication and to leverage browser caching of static files.

Code you enter here replaces the `#PAGE_JAVASCRIPT#` substitution string in the page template.

 **Tip:**

Do not to include opening or closing script tags. Just include the JavaScript code.

To reference a shortcut, use:

```
"SHORTCUTNAME"
```

- c. **Execute When Page Loads** - Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle Application Express.

 **Tip:**

To learn more about an attribute and view examples, select the attribute and click the **Help** tab in the central pane.

For example, adding the following to the Function and Global Variable Declaration attribute would test a function accessible from anywhere on the current page.

```
function test(){
    alert('This is a test.');
```

5. Click **Save**.

Configuring Rejoin Sessions

The Rejoin Sessions attribute controls support for joining existing sessions (similar to deep linking). Technically, this attribute determines whether URLs to this application contain session IDs.

Warning:

Enabling rejoin sessions may expose your application to possible security breaches, as it can enable attackers to take over existing end user sessions. To learn more, see "[About Rejoin Sessions](#)."

To use Rejoin Sessions, administrators must enable Rejoin Sessions at the instance-level. A more restrictive value of Rejoin Sessions at the instance-level overrides application and page settings.

To configure Rejoin Sessions in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, select the page name.
The Property Editor displays the Page attributes.
3. In the Property Editor, find the **Security** group.
4. Under Security, Rejoin Sessions, select an option:
 - **Application Default** - Inherits the setting defined at application-level.
 - **Disabled** - If the URL does not contain a session ID Application Express creates a new session.
 - **Enabled for Public Sessions** - If the URL goes to a public page and does not contain a session ID, Application Express attempts to utilize the existing session cookie established for that application. Application Express only joins using the cookie when the session is not yet authenticated.
Enabled for Public Sessions requires that Embed In Frames be set to **Allow from same origin** or **Deny**. This is not tied to a condition about the URL payload, but also applies to session state protected URLs.
 - **Enabled for All Sessions** - If the URL does not contain a session ID, Application Express attempts to utilize the existing session cookie established for that application, providing the following conditions are met:
 - Session State Protection is enabled for the application and the URL includes a valid checksum. For public bookmarks, the most restrictive item level protection must be either **Unrestricted** or **Checksum Required - Application Level**.

- The URL does not contain payload (a request parameter, clear cache or data value pairs).

This setting also requires that Embed In Frames is set to Allow from same origin or to **Deny** for the application.

 **Tip:**

The option **Application Default** inherits the setting defined at application-level. See "[Browser Security](#)."

5. Click **Save**.

 **See Also:**

- "[Browser Security](#) "
- "[About Rejoin Sessions](#)"
- "Configuring Rejoin Sessions for an Instance" in *Oracle Application Express Administration Guide*

Configuring Dialog Attributes

 **Tip:**

Creating a dialog page is a multiple step process. To learn more, see "[Creating Dialog Pages](#)."

To configure Dialog attributes in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, select the page name.
The Property Editor displays the Page attributes.
3. In the Property Editor, find **Appearance** and verify that the Page Mode is set to either **Modal Dialog** or **Non-Modal Dialog**.
4. Expand **Dialog** and configure these attributes:

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- a. Width
 - b. Height
 - c. Maximum Width
 - d. Attributes
 - e. CSS Classes
 - f. Chained
5. To save your changes, click **Save** or **Save and Run Page**.

 **See Also:**

"[Creating Dialog Pages](#)" and [Right Pane of Page Designer \(Property Editor\)](#)

Configuring Page Attributes to Warn Users of Unsaved Changes

Developers can enable the **Warn on Unsaved Changes** page attribute to warn users of unsaved changes on a page when they attempt to navigate away from it.

To enable the Warn on Unsaved Changes page attribute:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, select the page name.
The Property Editor displays the Page attributes.
3. In the Property Editor, find the **Navigation** group.
4. For **Warn on Unsaved Changes**, select **Yes** to warn users.
5. Click **Save**.

 **Tip:**

The **Warn on Unsaved Changes** attribute is implemented using the `apex.page.warnOnUnsavedChanges` API.

 **See Also:**

- ["Configuring Button Attributes to Warn Users of Unsaved Changes"](#)
- ["Configuring Item Attributes to Warn Users of Unsaved Changes"](#)
- [Oracle Application Express API Reference](#)

Running a Page from Page Designer

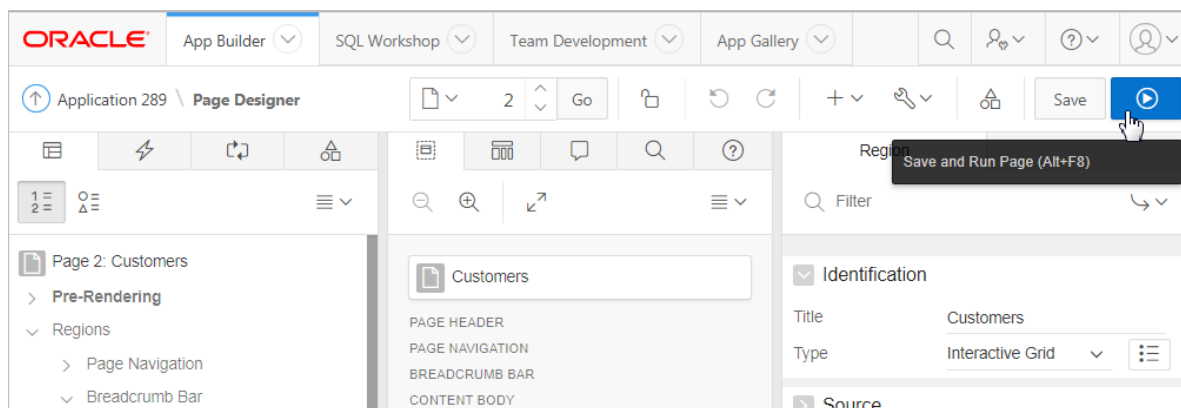
The Application Express engine dynamically renders and processes pages based on data stored in database tables. To view a rendered version of an application, you run or submit it to the Application Express engine.

 **Tip:**

You cannot run Modal and Non-Modal dialog pages directly using the Save and Run Page button. To view a Modal and Non-Modal dialog page, run the normal page that is responsible for opening the dialog page and access it from there.

To run a page in Page Designer:

1. View the page in Page Designer. See ["Viewing a Page in Page Designer."](#)
You preview a page by running it. Running an application page displays the rendered version as end users would see it.
2. Click the **Save and Run Page** button in the upper right corner. The Save and Run Page button resembles a blue right-facing arrow.



A rendered version of the application page appears.

**See Also:**

"[Running an Application or Page](#)" and "[Runtime Developer Toolbar](#)"

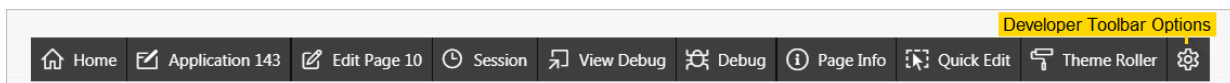
Runtime Developer Toolbar

Use the Developer toolbar to quickly edit the current application or currently running page, or view session state.

When you run an application from App Builder and view a page, the Runtime Developer toolbar displays at the bottom of any editable running page.

**Tip:**

The Runtime Developer toolbar only displays if the Availability, Status attribute on the Application Definition page is set to **Available with Developer Toolbar**

**Note:**

The Runtime Developer toolbar does not display on locked packaged applications pages or on jQuery Mobile Smartphone pages.

The Runtime Developer toolbar contains the following controls:

- **Home** links to the Workspace home page. See "[Understanding the Workspace Home Page](#)."
- **Application** links to the Application home page. See "[Understanding the App Builder Home Page](#)."
- **Edit Page** displays the currently running page ID.
- **Session** links you to session state information for the current page. See "[Viewing Session State](#)."
- **View Debug** displays the Debug reports. See "[Viewing Session State](#)."
- **Debug** toggles the page between **Debug** and **No Debug** mode. See "[Utilizing Debug Mode](#)."
- **Page Info:**
 - **Show Layout Columns** toggles between **Show Layout Columns** and **Hide Layout Columns**. This option only applies if a layout is used and that layout supports showing layout columns and the page has more than one column.

- **Show Page Timing** displays the Page Performance Timing dialog. Click **Copy** to copy the data in table form and then paste it into another application. Click **Clear** to remove the current timing events.
 - **Quick Edit** enables developers to enter either Quick Edit mode or access Live Template Options.
 - **Access Quick Edit Mode** - Click **Quick Edit** and then select the desired component to instantly access the component in Page Designer. Press `ESCAPE` or click outside a component to exit quick edit mode.
 - **Edit Live Template Options** - Click **Quick Edit** and then move the mouse over the component for which you want to modify template options and click the Wrench icon in the upper right corner. See "[Using Quick Edit to Modify Live Template Options.](#)"
- Theme Roller** enables developers to easily customize the appearance of an application by selecting colors from color picker and setting values. Only displays for themes supporting Theme Roller. See "[Using Theme Styles and Theme Roller.](#)"
- **Developer Toolbar Options** displays on the far right and resembles a mechanical gear. Clicking Developer Toolbar Options to customize how the Runtime Developer Toolbar displays. Available controls include:
 - **Auto Hide**
 - **Show Icons Only**
 - **Display Position** (Top, Left, Bottom, Right)

About JavaScript Error Detection

If a page has one or more JavaScript errors, a red error button displays on the left side of the Runtime Developer toolbar. If you have enabled **Auto Hide**, the Runtime Developer toolbar displays indicating the error.

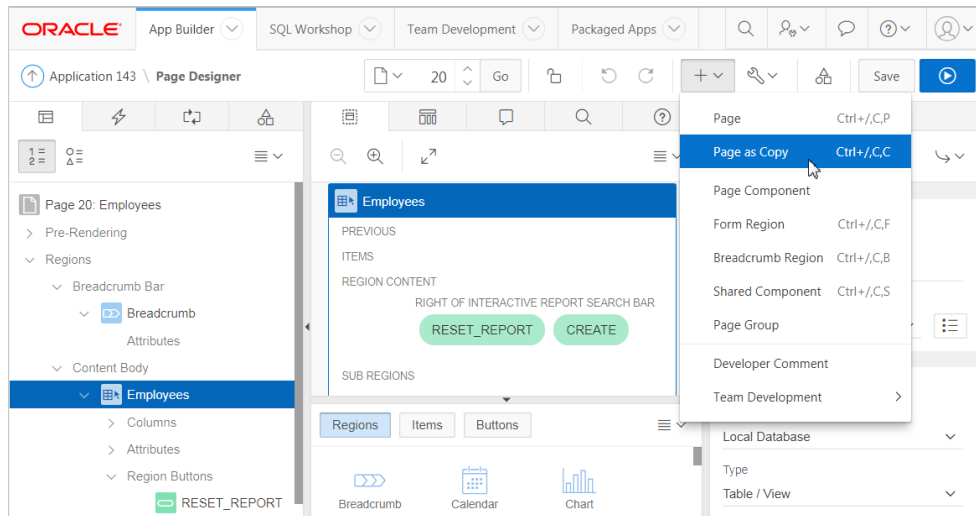
Click the red error button to view an alert dialog which explains you should view the Browser console. The red error button persists until error has been resolved, the Browser console is cleared, and you refresh the page.

Copying a Database Application Page

You can copy a page from the current application or from another application. During the copy process, you can also copy shared components or change mappings to shared components in the target application.

To copy a page:

1. View the page you want to copy in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. On the Page Designer toolbar, click the **Create** menu and then select **Page as Copy**.



3. On Copy Page Option, select one of the following and click **Next**:
 - **Page in this application**
 - **Page in another application**
4. Follow the on-screen instructions.

Deleting Pages

Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries.

- [Deleting a Page](#)
- [Deleting Multiple Pages](#)
- [Deleting a Range of Pages](#)



See Also:

["Locking and Unlocking a Page"](#)

Deleting a Page

To delete a page in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
 Page Designer appears.
2. Verify the page number.
3. On the Page Designer toolbar, select the **Utilities** menu and click **Delete Page**.
The Confirm Page Delete page appears.

4. Click **Permanently Delete Page**.



See Also:

"Page Designer Toolbar"

Deleting Multiple Pages

To delete multiple pages:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities, click **Cross Page Utilities**.
5. Click **Delete Multiple Pages**.
6. Select the pages to be deleted and click **Delete Pages**.

Deleting a Range of Pages

To delete a range of pages:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities, click **Cross Page Utilities**.
5. Click **Delete Multiple Pages**.
6. Click the **Delete Pages by Range** tab.
7. Enter the pages to be deleted and click **Delete Pages**.

9

Developing Reports

In Oracle Application Express, a report is the formatted result of a SQL query. You can generate the SQL query by selecting a table or view in a wizard, or by defining the SQL query manually.

- [Understanding Report Types](#)
When creating a database application, developers can create four basic report types: interactive grid, interactive report, report on form on table, or a classic report.
- [Creating a Report Using the Create Application Wizard](#)
Run the Create Application Wizard to create a new application containing one or more pages. Supported reports include interactive report, classic report, and interactive grid.
- [Creating a Report Using the Create Page Wizard](#)
Run the Create Page Wizard from an existing application to create a variety of different reports. Supported reports include interactive report, interactive grid, classic report, report with form on table, list view, column toggle report, reflow report, or report on a web service.
- [Creating a Report by Dragging and Dropping from the Gallery](#)
Create a report by dragging and dropping a report type from the Gallery.
- [Managing Interactive Grids](#)
An interactive grid presents users a set of data in a searchable and customizable report. Developers control how an interactive grid works by editing the following attributes in Page Designer: region attributes, report Attributes, and Column attributes.
- [Managing Interactive Reports](#)
An interactive report is the formatted result of a SQL query that is only supported on Desktop pages. End users can customize the report layout and data displayed by selecting options on the Actions menu. Developers control how a interactive report works by editing the following attributes in Page Designer: region attributes, report Attributes, and Column attributes.
- [Managing Classic Reports](#)
A classic report is the formatted result of a SQL query. Developers choose a table on which to build a report, or provide a custom SQL SELECT statement or a PL/SQL function returning a SQL SELECT statement. Developers control how a classic report works by editing the following attributes in Page Designer: region attributes, report Attributes, and Column attributes.
- [Printing Report Regions](#)
Configure a report region to print by exporting it to several different formats.
- [Understanding BLOB Support in Forms and Reports](#)
Oracle Application Express includes declarative BLOB support to enable developers to declaratively upload files in forms, and download or display files in reports.

**See Also:**

["Developing Forms"](#) and ["Managing Database Application Components"](#)

Understanding Report Types

When creating a database application, developers can create four basic report types: interactive grid, interactive report, report on form on table, or a classic report.

The main difference between these different report types is the extent and way in which end users can customize the appearance of the data through searching, filtering, sorting, column selection, highlighting, and other data manipulations.

**Tip:**

When creating reports for mobile devices, Oracle recommends creating components that are optimized to work well on mobile devices. To learn more, see ["Creating Applications for Mobile Devices."](#)

- [Interactive Grid](#)
- [Interactive Report](#)
- [Classic Report](#)
- [Report and Form](#)

Interactive Grid

An interactive grid presents users a set of data in a searchable, customizable report. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse and keyboard.

Users can lock, hide, filter, freeze, highlight, sort individual columns, and create control breaks on specific columns using the **Actions** and **Column Heading** menus. Advanced users can also define breaks and aggregations against columns. Users can also directly customize the appearance of an interactive grid. Users can use the mouse and keyboard to resize the width of a column and drag and drop columns into different places in the grid. Once customized, the report can be saved as either a private or public report. Both the Create Application Wizard and Create Page Wizard support the creation of interactive grids.

The following is an example of an interactive grid.

| Status Id | Name | Description | Project Lead | Completed | Created | Updated |
|-----------|-----------------------------------|--------------------------|-------------------|-------------|-------------|-------------|
| 3 | Configure Web Development To... | Deter... | Lucille Beatie | 03-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Train Developers on Web develo... | Ensuri... | Lucille Beatie | 14-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Migrate Legacy Applications | Move... | Miyazaki Yokoh... | - | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Develop Partner Portal POC | Develo... | Bernard Jackman | - | 01-MAR-2018 | 01-MAR-2018 |
| 1 | Develop Production Partner Portal | Develo... | Lucille Beatie | - | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop New Reporting Apps | Develo... | Lucille Beatie | 28-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop IT Management Apps | Develo... | Bernard Jackman | 08-JAN-2018 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop Customer Tracker Appli... | Develo... | Lucille Beatie | 28-JAN-2018 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Implement Customer Satisfactio... | Imple... | Bernard Jackman | 28-JAN-2018 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Improve IT Management Apps | Enhance apps to allow... | Bernard Jackman | 25-FEB-2018 | 01-MAR-2018 | 01-MAR-2018 |

Editable Interactive Grid

Developers have the option of making an interactive grid editable. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page. Editable interactive grids include additional controls. A Row Actions menu displays at the start of each row and enables users to add, edit, and refresh rows. Edit, Save, and Add Row buttons also display to the right of the Actions menu.

The following is an example of an editable interactive grid.

| Status Id | Name | Description | Project Lead | Completed | Created | Updated |
|-----------|-----------------------------------|-------------------------|-------------------|-------------|-------------|-------------|
| 3 | Configure Web Development ... | Determine the hard... | Lucille Beatie | 03-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Train Developers on Web develo... | Ensure all develop... | Lucille Beatie | 14-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Migrate Legacy Applications | Move the data and r... | Miyazaki Yokoh... | - | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Develop Partner Portal POC | Develop a proof of c... | Bernard Jackman | - | 01-MAR-2018 | 01-MAR-2018 |
| 1 | Develop Production Partner P... | Develop the product... | Lucille Beatie | - | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop New Reporting Apps | Develop apps to me... | Lucille Beatie | 28-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop IT Management Apps | Develop apps to allo... | Bernard Jackman | 08-JAN-2018 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop Customer Tracker Ap... | Develop an applicati... | Lucille Beatie | 28-JAN-2018 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Implement Customer Satisfact... | Implement an applic... | Bernard Jackman | 28-JAN-2018 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Improve IT Management Apps | Enhance apps to all... | Bernard Jackman | 25-FEB-2018 | 01-MAR-2018 | 01-MAR-2018 |

See Also:

- *Oracle Application Express End User's Guide*
- "Managing Interactive Grids"
- "Making an Existing Interactive Grid Editable"

Interactive Report

An interactive report is a formatted result of a SQL query. Both the Create Application Wizard and Create Page Wizard support the creation of interactive reports. You choose a table on which to build a report or provide a custom SQL SELECT statement. Interactive reports are only supported for Desktop applications. End users can customize the report layout and data displayed by selecting options on the Actions menu.

The following is an example of an interactive report.

| Status Id | Name | Description | Project Lead | Completed | Created | Updated |
|-----------|--|---|-------------------|-------------|-------------|-------------|
| 3 | Configure Web Development Tool Environment | Determine the hardware and software required to develop web development tool. | Lucille Beatie | 03-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Train Developers on Web development tool | Ensure all developers developing with the new appropriate training. | Lucille Beatie | 14-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Migrate Legacy Applications | Move the data and re applications currently legacy servers | Miyazaki Yokohama | - | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Develop Partner Portal POC | Develop a proof of co can use to work more us. | Bernard Jackman | - | 01-MAR-2018 | 01-MAR-2018 |
| 1 | Develop Production Partner Portal | Develop the productio can use to work more us. | Lucille Beatie | - | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop New Reporting Apps | Develop apps to meet C Level reporting requirements. | Lucille Beatie | 28-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop IT Management Apps | Develop apps to allow IT to manage resources. | Bernard Jackman | 08-JAN-2018 | 01-MAR-2018 | 01-MAR-2018 |

Developers can include multiple interactive reports per page and can restrict the capabilities available to end users (such as disabling download or support for hiding column). When the end user views the report, report functionality is same across all reports in the application.

When viewing an interactive report, end users can customize how and what data displays. By default, interactive reports include a search bar, an Actions menu, column heading menus, and Edit icons in the first column of each row. Using options on the Actions menu, users can alter the report layout by hiding or exposing specific columns and applying filters, highlighting, and sorting. They can also define breaks, aggregations, charts, group bys, and add their own computations. Once customized, the report can be saved as either a private or public report.

 **See Also:**

- ["Managing Interactive Reports "](#)
- ["Using Interactive Reports" in Oracle Application Express End User's Guide](#)

Classic Report

A classic report is a formatted result of a SQL query. You choose a table on which to build a report, or provide a custom SQL SELECT statement or a PL/SQL function returning a SQL SELECT statement. With the exception of sorting and simple filtering, end users cannot customize a classic report. Both the Create Application Wizard and Create Page Wizard support the creation of classic reports.

The following is an example of a classic report.

| Status Id | Name | Description | Project Lead | Budget | Completed | Created | Updated |
|-----------|--|--|-------------------|--------|-------------|-------------|-------------|
| 3 | Configure Web Development Tool Environment | Determine the hardware and software required to develop with Web development tool. | Lucille Beatie | 5000 | 03-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Train Developers on Web development tool | Ensure all developers who will be developing with the new tool get the appropriate training. | Lucille Beatie | 20000 | 14-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Migrate Legacy Applications | Move the data and redevelop the applications currently running on top of legacy servers | Miyazaki Yokohama | 38000 | | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Develop Partner Portal POC | Develop a proof of concept that partners can use to work more collaboratively with us. | Bernard Jackman | 25000 | | 01-MAR-2018 | 01-MAR-2018 |
| 1 | Develop Production Partner Portal | Develop the production app that partners can use to work more collaboratively with us. | Lucille Beatie | 85000 | | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop New Reporting Apps | Develop apps to meet C Level reporting requirements. | Lucille Beatie | 15000 | 28-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |

To create a classic report, developers either select a table or provide a SQL statement. Classic reports support general keyword search capability, the ability to specify the number of rows that display, and basic column sorting.

 **Tip:**

Developers can create highly customized reports using report template (in particular, named column templates). See ["Report Templates."](#)

 **See Also:**

["Managing Classic Reports"](#)

Report and Form

Developers can create a report and form on a table. Developers select a Report Type to determine if the report is an interactive grid, interactive report, or a classic report. Users click an Edit icon to access the form.

Both the Create Application Wizard and Create Page Wizard support the creation of a report and form combination. The main difference between the Create Application Wizard and Create Page Wizard is the amount of customization. With the Create Page Wizard the developer can select the report type (that is, interactive grid, interactive report, or classic) and the table. Additionally, the developer can also specify whether to include and configure breadcrumbs or a navigation menu and select the columns and the order in which they display.

The following is an example of an interactive grid report and form.

| Status Id | Name | Description | Project Lead | Completed | Created | Updated |
|-----------|--|--|-------------------|-------------|-------------|-------------|
| 3 | Configure Web Development Tool Environment | Determine t... | Lucille Beatie | 03-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Train Developers on Web development tool | Ensure all d... | Lucille Beatie | 14-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Migrate Legacy Applications | Move the d... | Miyazaki Yokohama | - | 01-MAR-2018 | 01-MAR-2018 |
| 2 | Develop Partner Portal POC | Develop a p... | Bernard Jackman | - | 01-MAR-2018 | 01-MAR-2018 |
| 1 | Develop Production Partner Portal | Develop the... | Lucille Beatie | - | 01-MAR-2018 | 01-MAR-2018 |
| 3 | Develop New Reporting Apps | Develop apps to meet C Level reporting requirements. | Lucille Beatie | 28-DEC-2017 | 01-MAR-2018 | 01-MAR-2018 |

See Also:

- ["Interactive Grid"](#)
- ["Interactive Report"](#)
- ["Classic Report"](#)

Creating a Report Using the Create Application Wizard

Run the Create Application Wizard to create a new application containing one or more pages. Supported reports include interactive report, classic report, and interactive grid.

- [Creating an Interactive or Classic Report Using the Create Application Wizard](#)
- [Creating an Interactive Grid Using the Create Application Wizard](#)

- [Report Options When Running the Create Application Wizard](#)

Creating an Interactive or Classic Report Using the Create Application Wizard

Run the Create Application Wizard, you can create a new application that contains one or multiple interactive report or classic report pages.

To create a report using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Click **New Application**.
4. For Name, enter the name used to identify the application to developers.
5. For Appearance, accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
6. To add a report, click **Add Page** and select then select **Report**.
7. On Add Report Page:
 - a. Page Name - Enter a name for this page.
 - b. Set Icon - Select an icon to display in the navigation menu for this page.
 - c. Page Source - Select the source for the report:
 - **Table or View**
 - **SQL Query**

What you select determines what displays next. Follow the on-screen instructions. To learn more about an attribute, click the Help icon in the lower left corner.
 - d. Select a report type:
 - **Interactive Report**
 - **Classic Report**
 - e. Lookup Columns - Expand to see attributes for defining a lookup to another table. Use Lookup Columns to replace identifiers with a display column, such as showing the department name instead of the department number. Specify the following:
 - Lookup Key - Select the column to change to output the display column.
 - Display Column - Select the column (from the lookup table) to be displayed.

 **Tip:**

Lookup Columns only appear if the current page is based on a table and has foreign key constraints to another table.

- f. Advanced - Expand **Advanced** to configure these options:

- Set as Home Page - Enable this to make this page the home page for the application. Any page previously defined as the home page are updated.
- Set as Administration Page - Enable this to have this page display under Administration, rather than as a normal page in the application. If enabled, you access the page by clicking Administration in the main menu, and then selecting from the Application Administration list.
- Page Help - Enter text to be displayed when the user selects Page Help.

 **Tip:**

This setting requires you select the **About Page** feature. If you select the **About Page** feature, a help icon is generated in the navigation bar with an entry for page help.

- g.** Click **Add Page**.
- 8.** Under Features, select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.

 **Tip:**

Click the **Check All** button to select all features.

- 9.** Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
- 10.** Click **Create Application**.

 **See Also:**

- ["Understanding Page Types, Features, and Settings"](#)
- ["Report Options When Running the Create Application Wizard"](#)
- ["Managing Interactive Grids"](#)

Creating an Interactive Grid Using the Create Application Wizard

Run the Create Application Wizard, you can create a new application that contains one or multiple interactive grid report pages.

To create an interactive grid using the Create Application Wizard:

- 1.** On the Workspace home page, click the **App Builder** icon.
- 2.** Click the **Create** button.
- 3.** Click **New Application**.
- 4.** For Name, enter the name used to identify the application to developers.

5. For Appearance, accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
6. To add a report, click **Add Page** and select then select **Interactive Grid**.
7. On Add Interactive Grid Page:
 - a. Page Name - Enter a name for this page.
 - b. Set Icon - Select an icon to display in the navigation menu for this page.
 - c. Page Source - Select the source for the report:
 - **Table or View**
 - **SQL Query**

What you select determines what displays next. Follow the on-screen instructions. To learn more about an attribute, click the Help icon in the lower left corner.
 - d. Determine if the report is editable. Select either:
 - **Allow Editing**
 - **Read Only**
 - e. Advanced - Expand **Advanced** to configure these options:
 - Set as Home Page - Enable this to make this page the home page for the application. Any page previously defined as the home page are updated.
 - Set as Administration Page - Enable this to have this page display under Administration, rather than as a normal page in the application. If enabled, you access the page by clicking Administration in the main menu, and then selecting from the Application Administration list.
 - Page Help - Enter text to be displayed when the user selects Page Help.

 **Tip:**

If you enter text in **Page Help**, the wizard creates page help text. To enable end users to access that help, you must select the **About Page** feature. Then, the wizard creates a help icon that end users can select to access the Page Help.

- f. Click **Add Page**.
8. Under Features, select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.

 **Tip:**

Click the **Check All** button to select all features.

9. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
 10. Click **Create Application**.



See Also:

- ["Understanding Page Types, Features, and Settings"](#)
- ["Report Options When Running the Create Application Wizard"](#)
- ["Managing Interactive Grids"](#)

Report Options When Running the Create Application Wizard

Use the Create Application Wizard to create interactive reports and classic reports (with or without a from page for updating records) or an interactive grid.

Available Reports When Creating Applications

The Create Application Wizard supports the following report types.

Table 9-1 Create Application Wizard - Available Reports

| Report Type | Description | To Learn More |
|------------------|--|---|
| Report | <p>Creates a page that contains the formatted result of a SQL query. First, you select a page source (that is, Table or View or SQL Query). Second, you select a report type (that is, Interactive Report or Classic Report). To include a form page for creating or updating records, select Include Form.</p> <p>If the report is based on table which has a foreign key constraints to another table, the developer can also define Lookup Columns. Use Lookup Columns to replace identifiers with a display column, such as showing the department name instead of the department number.</p> | <p>See:</p> <ul style="list-style-type: none"> • "Creating an Interactive or Classic Report Using the Create Application Wizard" • "Managing Interactive Reports" • "Managing Classic Reports" |
| Interactive Grid | <p>An interactive grid presents users a set of data in a searchable, customizable report. First, you select a page source (that is, Table or View or SQL Query). Second, you determine if the interactive grid is editable by selecting Allow Editing or Read Only.</p> <p>If you select Allow Editing, users can add to, modify, and refresh the data set directly on the page. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard. You choose a table on which to build the interactive grid.</p> | <p>See:</p> <ul style="list-style-type: none"> • "Creating an Interactive Grid Using the Create Application Wizard" • "Managing Interactive Grids" |

 **See Also:**

["Creating an Interactive or Classic Report Using the Create Application Wizard"](#) and ["Available Page Types in the Create Application Wizard"](#)

Creating a Report Using the Create Page Wizard

Run the Create Page Wizard from an existing application to create a variety of different reports. Supported reports include interactive report, interactive grid, classic report, report with form on table, list view, column toggle report, reflow report, or report on a web service.

- [Creating a Report on a New Page](#)
Use the Create Page Wizard to wide variety of reports including interactive report, interactive grid, classic report, report with form on table, list view, column toggle report, reflow report, and report on web service.
- [Creating a Classic or Interactive Report on New Page Using Remote Database References](#)
Run the Create Page Wizard to add a new page with a classic or interactive report that uses a remote database reference.
- [Report Options When Running the Create Page Wizard](#)
Learn about report options when running the Create Page Wizard.

Creating a Report on a New Page

Use the Create Page Wizard to wide variety of reports including interactive report, interactive grid, classic report, report with form on table, list view, column toggle report, reflow report, and report on web service.

 **Tip:**

This section describes how to create a report by running the Create Page Wizard from the Application home page. You can also run this wizard in Page Designer by clicking the **Create** menu and selecting **Page**.

To create a report on a new page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
The Application home page appears.
3. Click the **Create Page** button.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.

- b. Page Type - To create a report, select **Component** and then **Report**.

 **Tip:**

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

- c. Select a report type. Options include:

- **Interactive Report**
- **Interactive Grid**
- **Classic Report**
- **Report with Form on Table**
- **List View** (Optimized for mobile apps)
- **Column Toggle Report** (Optimized for mobile apps)
- **Reflow Report** (Optimized for mobile apps)
- **Report on Web Service**

 **Tip:**

To learn more about each report type, see "[Report Options When Running the Create Page Wizard](#)."

- d. Click **Next**.
5. What appears next depends upon the selected report type. Follow the on-screen instructions.
- To learn more about an attribute, see field-level Help.
6. Click **Create**.

Creating a Classic or Interactive Report on New Page Using Remote Database References

Run the Create Page Wizard to add a new page with a classic or interactive report that uses a remote database reference.

 **Tip:**

This section describes how to create a report by running the Create Page Wizard from the Application home page. You can also run this wizard in Page Designer by clicking the **Create** menu and selecting **Page**.

To add a classic report or interactive report on a new page:

1. Create REST Enabled SQL reference.
2. On the Workspace home page, click the **App Builder** icon.
3. Select the application.
4. Click **Create Page**.
5. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Report**.

 **Tip:**

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

6. On Create Page:
 - a. Select either **Interactive Report** or **Classic Report**.
 - b. Click **Next**
7. For Page Attributes:
 - a. Page Number - Enter a page number. If you identify a new page number, the wizard creates a new page. If you identify an existing page number, the wizard adds the component to that page.
 - b. Page Name - Specify a name for the page.
 - c. Page Mode - Identify the page mode.
To learn more, see field-level Help.
 - d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - e. Click **Next**.
8. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
9. On Report Source, select a Data Source. Options include
 - **Local Database** - Data is sourced from a local database.
 - **REST Enabled SQL Service** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL Reference.
 - **Web Source** - Data is sourced from a RESTful web service defined using Web Source Modules.

Select either **REST Enabled SQL Service** or **Web Source**.

What displays next changes based on your selection. Follow the on-screen instruction. To learn more about an attribute, see field-level Help.

10. Click **Create**.



See Also:

- ["Creating a REST Enabled SQL Service Reference"](#)
- ["Managing Web Source Modules"](#)

Report Options When Running the Create Page Wizard

Learn about report options when running the Create Page Wizard.



Tip:

Universal Theme - 42 is optimized to work equally well in either a mobile or desktop environment. In previous releases and in earlier themes, the Mobile user interface is based on jQuery Mobile. Oracle recommends migrating existing mobile applications to the Universal Theme as soon as possible.

Available Reports in the Create Page Wizard

The Create Page Wizard supports the following reports when running the Create Page Wizard with the Universal Theme.

Table 9-2 Create Page Wizard — Available Reports

| Report Type | Description | To Learn More |
|--------------------|--|--|
| Interactive Report | <p>An interactive report is a formatted result of a SQL query. Developers choose a table on which to build a report, or provide a custom SQL SELECT statement. Developers also select the columns to display in the report.</p> <p>If the report is based on table which has a foreign key constraints to another table, the developer can also define Lookup Columns. Use Lookup Columns to replace identifiers with a display column, such as showing the department name instead of the department number.</p> <p>End users can customize the report layout and data displayed by selecting options on the Actions menu.</p> | See " Managing Interactive Reports " |

Table 9-2 (Cont.) Create Page Wizard — Available Reports

| Report Type | Description | To Learn More |
|---------------------------|---|--|
| Interactive Grid | <p>An interactive grid is a formatted result of a SQL query. Developers choose a table on which to build the report, or provide a custom SQL query.</p> <p>End users can alter the report layout using Column menus and sort options, rearrange columns by dragging and dropping, and change how data displays using options on the Actions menu.</p> <p>Interactive grids can be editable or non-editable. When an interactive grid is editable, end users can edit the underlying data, add rows, and delete rows. To create an editable interactive grid when running the Create Page Wizard, selecting Yes for the Editing Enabled option.</p> | See " Managing Interactive Grids " or " Making an Existing Interactive Grid Editable " |
| Classic Report | <p>Formatted result of a SQL query. Developers choose a table on which to build a report, or provide a custom SQL SELECT statement. Developers also select the columns to display in the report.</p> <p>If the report is based on table which has a foreign key constraints to another table, the developer can also define Lookup Columns. Use Lookup Columns to replace identifiers with a display column, such as showing the department name instead of the department number.</p> <p>With the exception of sorting and simple filtering, end users cannot customize a classic report.</p> | See Managing Classic Reports |
| Report with Form on Table | Creates an interactive grid report and form based on the table you specify. For the report, you choose a report type (Interactive Report , Interactive Grid , or Classic Report). | Not applicable. |
| List View | <p>Optimized for mobile applications.</p> <p>Features a responsive design to display data and provide easy navigation on Smartphones. Creates a page that contains the formatted result of a SQL query. You choose a table on which to build the List view and select a database column to be used for the List view entry.</p> | Not applicable. |
| Column Toggle Report | <p>Optimized for mobile applications.</p> <p>Creates a responsive report designed for mobile applications and Smartphones. By default, column toggle reports are created with all columns set to the same priority. However, the developer can edit the report column attributes and rank columns by importance. Columns with a lesser priority (larger number) are hidden at narrower screen widths. The report includes a Columns button which enables end users to select which columns they want to view.</p> | Not applicable. |

Table 9-2 (Cont.) Create Page Wizard — Available Reports

| Report Type | Description | To Learn More |
|-----------------------|---|---|
| Reflow Report | Optimized for mobile applications. Creates a responsive report designed for mobile applications and Smartphones. When there is not enough space available to display the report horizontally, the report responds by collapsing the table columns into a vertical value pairs layout where each column displays on a separate row. | See " Developing Reports " |
| Report on Web Service | Creates a report on a Web Service result. | "See Managing Legacy Web Services " |

Available Reports for Mobile User Interface Applications

In previous releases, developers selected the **Mobile** User Interface to optimize applications for mobile environments. The Mobile user interface is based on jQuery Mobile. If your application users an older theme and the Mobile User Interface, the Create Page Wizard supports the following reports.

Table 9-3 Create Page Wizard — Supported Reports for Mobile User Interface Applications

| Page Type | Description | To Learn More |
|-----------------------|---|--|
| List View | Optimized for mobile applications. Features a responsive design to display data and provide easy navigation on Smartphones. Creates a page that contains the formatted result of a SQL query. You choose a table on which to build the List view and select a database column to be used for the List view entry. | See " Developing Reports " |
| Column Toggle Report | Optimized for mobile applications. Creates a responsive report designed for mobile applications and Smartphones. By default, column toggle reports are created with all columns set to the same priority. However, the developer can edit the report column attributes and rank columns by importance. Columns with a lesser priority displays at narrower widths. The report includes a Columns button which enables end users to select which columns they want to view. | See " Developing Reports " |
| Reflow Report | Optimized for mobile applications. Creates a responsive report designed for mobile applications and Smartphones. When there is not enough space available to display the report horizontally, the report responds by collapsing the table columns into a vertical value pairs layout where each column displays on a separate row. | See " Developing Reports " |
| Report on Web Service | Creates a report on a Web Service result. | See " Managing Legacy Web Services " |

Creating a Report by Dragging and Dropping from the Gallery

Create a report by dragging and dropping a report type from the Gallery.

 **Tip:**

When you drag and drop a report region from the Gallery, you must provide the underlying SQL query.

To add a report by dragging and dropping from the Gallery:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Gallery at the bottom of the central pane, locate the report type you want to create.

The Gallery lists all controls or components you can add to a page. Passing the cursor over a control or component displays a tooltip that describes it.

 **Tip:**

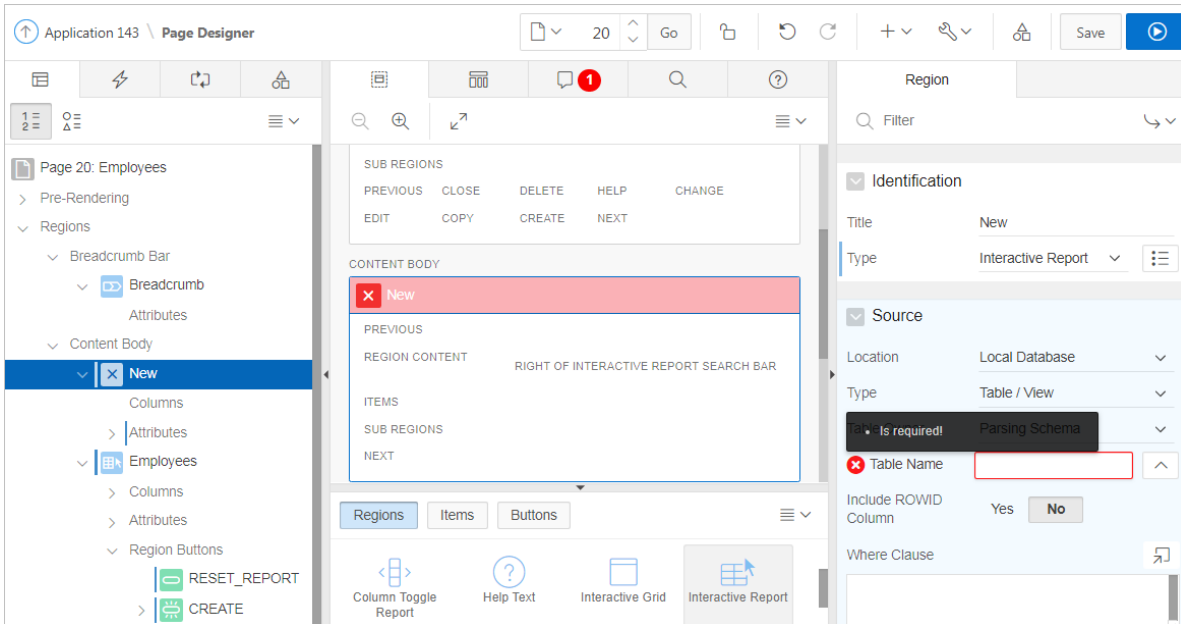
In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Select **Add To** and then the location where you want to add the component.

3. Click and hold the mouse on the component to be created and drag it the desired location in the Layout tab.

When the mouse is over the appropriate location, the Layout tab displays as a darkened yellow tile. Release the mouse to drop the component. You can only drop components into appropriate drop positions, as determined by the component type.

Based on the type of component you add, Page Designer indicates what actions are required next. The Messages tab displays a red or yellow badge indicating the number of messages you need to address.

The following example shows a new Interactive Report region with an error message indicating that a SQL statement is required.



4. Edit the appropriate attributes in the Property Editor.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

See Also:

"[Messages Tab](#)"

Managing Interactive Grids

An interactive grid presents users a set of data in a searchable and customizable report. Developers control how an interactive grid works by editing the following attributes in Page Designer: region attributes, report Attributes, and Column attributes.

Note:

To learn more about features and capabilities of interactive grids, install the sample application, *Sample Interactive Grids*. See "[Installing a Productivity and Sample App](#)."

- [Managing Interactive Grid Region Attributes](#)
Each application page contains one or more regions. A region is an area on a page that serves as a container for content. Developers edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.
- [Managing Interactive Grid Attributes](#)
Report Attributes control how an interactive grid works. Developers edit interactive grid Attributes to control if end users can edit the underlying data, configure report pagination, create error messages, configure the toolbar and download options, control if users can save public reports, and add Icon and Detail Views.
- [Managing Interactive Grid Column Attributes](#)
Column attributes enable developers to control the display, features, and behavior of interactive grid columns. Developers edit Column attributes to alter nearly all aspects of column behavior, including altering the layout and appearance, creating validations, defining column links, creating column filters, and adding support for export and printing.
- [About Customizing Interactive Grids in a Running Application](#)
When running an application, users can alter the report layout using Column menus and sort options, rearrange columns by dragging and dropping, and change how data displays using options on the Actions menu.
- [Saving Interactive Grids](#)
After users customize an interactive grid they can save it as either a Private or Public report.

 **See Also:**

- ["Interactive Grid"](#)
- *Oracle Application Express End User's Guide*
- ["Making an Existing Interactive Grid Editable"](#)

Managing Interactive Grid Region Attributes

Each application page contains one or more regions. A region is an area on a page that serves as a container for content. Developers edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.

- [Editing Interactive Grid Region Attributes](#)
- [Creating Master Detail from an Existing Interactive Grid](#)
- [Displaying Interactive Grids Conditionally](#)

 **See Also:**

- ["Editing Pages in Page Designer"](#)

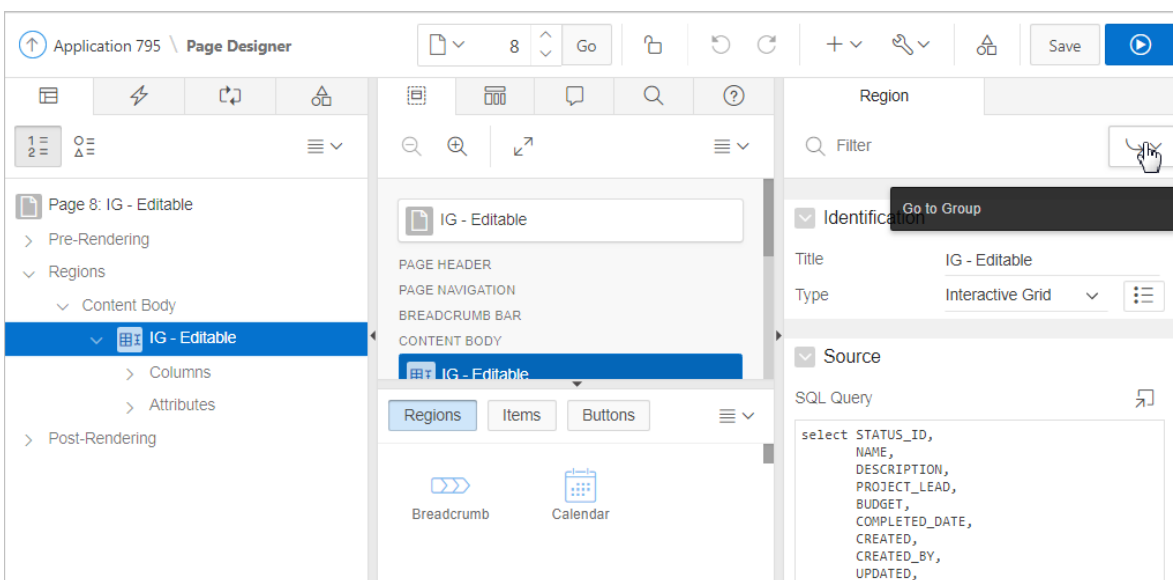
Editing Interactive Grid Region Attributes

To edit region attributes in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In the Rendering tab, select the region.

The Property Editor displays the region attributes in the right pane.



Use region attributes to alter the SQL source, change the region layout and appearance, create a master detail, define a region displays selector, or create conditions.

3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**See Also:**

"Right Pane of Page Designer (Property Editor)"

Creating Master Detail from an Existing Interactive Grid

A single page master detail features two editable interactive grids based on two related tables or views. You can define any number of master detail relationships on a single page by simply adding a related interactive grid to the page. If you have an existing interactive grid, you can add another related interactive grid by doing the following:

- Create the parent interactive grid.
- Define the primary key (if needed).
- Create the child interactive grid.
- For the child interactive grid, specify the Master Region and Master Column.

The following example demonstrates how to create an interactive grid on the departments table, `OEHR_DEPARTMENTS` by running the Create Page Wizard. This wizard creates the parent interactive grid and enables you to define the primary key. Then, you create a child interactive grid on the employees table, `OEHR_EMPLOYEES`.

To add a related interactive grid below an existing grid:

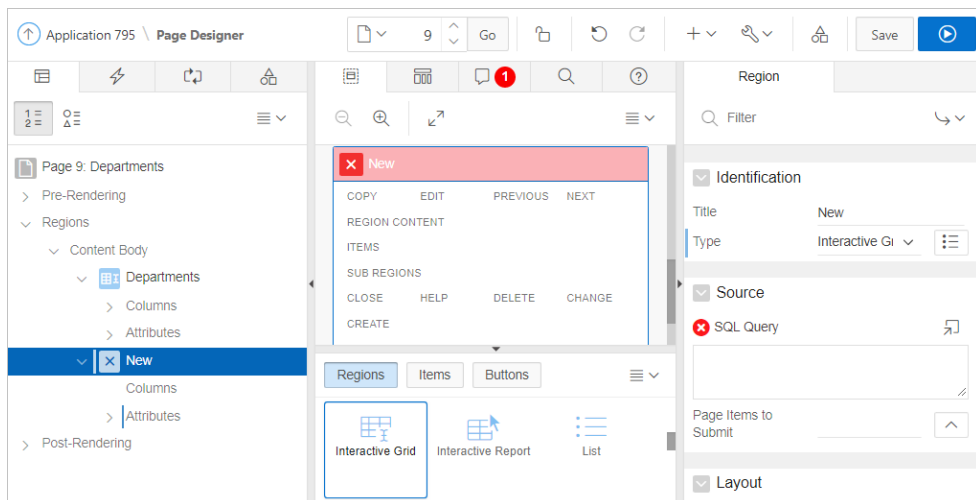
1. Create an application.
2. Run the Create Page Wizard. On the Application home page, click **Create Page**
3. In the Create Page Wizard:
 - a. Select a Page Type - Select **Report** and then select **Interactive Grid**.
 - b. Page Attributes - For Page Name enter `Departments` and click **Next**
 - c. Navigation Menu - Select how you want this page integrated into the Navigation Menu and click **Next**.
 - d. On Report Source:
 - Editing Enabled - Select **Yes**.
 - Source Type - Select **Table**.
 - Table / View Name - Select **OEHR_DEPARTMENTS (table)**.
 - Primary Key Column - Select **DEPARTMENT_ID (Number)**.
 - Click **Create**.Page Designer appears.
4. Click **Save and Run Page** to view a rendered version of the parent interactive grid.

This following illustration shows an editable interactive grid created on the `OEHR_DEPARTMENTS` table.

| | Department Name | Manager Id |
|-------------------------------------|------------------|------------|
| <input checked="" type="checkbox"/> | Administration | 200 |
| <input type="checkbox"/> | Marketing | 201 |
| <input type="checkbox"/> | Purchasing | 114 |
| <input type="checkbox"/> | Human Resources | 203 |
| <input type="checkbox"/> | Shipping | 121 |
| <input type="checkbox"/> | IT | 103 |
| <input type="checkbox"/> | Public Relations | 204 |

5. Return to Page Designer. Click **Edit Page X** on the Runtime Developer toolbar.
6. Create the child interactive grid on the OEHR_EMPLOYEES table:
 - a. In the Gallery, locate the Regions tab.
 - b. Under Regions, find and then right-click **Interactive Grid**.
 - c. Select **Add To, Content Body**, the region, **Departments**, and then **After**.

The Rendering tab and Layout tab both display a a new region in red.



7. Edit the New child region:
 - a. In the Rendering tab, select the New region (if not already selected).
 - b. In the Property Editor, edit the region attributes:

Note:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.

- Identification, Title - Enter `Employees`.

- Source, SQL Query - Enter the query:

```
select OEHR_EMPLOYEES.EMPLOYEE_ID as EMPLOYEE_ID,
       OEHR_EMPLOYEES.FIRST_NAME as FIRST_NAME,
       OEHR_EMPLOYEES.LAST_NAME as LAST_NAME,
       OEHR_EMPLOYEES.HIRE_DATE as HIRE_DATE,
       OEHR_EMPLOYEES.EMAIL as EMAIL,
       OEHR_EMPLOYEES.PHONE_NUMBER as PHONE_NUMBER,
       OEHR_EMPLOYEES.JOB_ID as JOB_ID,
       OEHR_EMPLOYEES.SALARY as SALARY,
       OEHR_EMPLOYEES.DEPARTMENT_ID as DEPARTMENT_ID
from OEHR_EMPLOYEES
```

- Layout, Start New Row - Select **Yes**.
 - Appearance, Template - Select **Standard**
 - Master Detail, Master Region - Select the region, **Departments**.
8. For the child interactive grid, Employees, define DEPARTMENTS as the Master Column:
 - a. In the Rendering tab, under the child interactive grid, **Employees**, expand the Columns node and select **DEPARTMENT_ID**.
 - b. In Property Editor, edit the following:
 - Find Master Detail.
 - For Master Column, select **DEPARTMENT_ID**.
 9. Click **Save**. To run the page, click **Save and Run Page**.

The screenshot shows two interactive grids. The top grid is the parent, with columns 'Department Name' and 'Manager Id'. The 'Accounting' department is selected, and its 'Manager Id' is 205. The bottom grid is the child, titled 'Employees', which displays employee details filtered by the selected department. The columns are 'Employee Id', 'First Name', 'Last Name', 'Hire Date', 'Email', 'Phone Num...', 'Job Id', and 'Salary'. Two employees are shown: Shelley Higgins (Employee Id 205, Job Id AC_MGR, Salary 12000) and William Gietz (Employee Id 206, Job Id AC_ACCOUNT, Salary 8300).

| Department Name | Manager Id |
|-----------------|------------|
| Construction | - |
| Benefits | - |
| Administration | 200 |
| Accounting | 205 |

| Employee Id | First Name | Last Name | Hire Date | Email | Phone Num... | Job Id | Salary |
|-------------|------------|-----------|-------------|----------|--------------|------------|--------|
| 205 | Shelley | Higgins | 07-FEB-2012 | SHIGGINS | 515.123.8080 | AC_MGR | 12000 |
| 206 | William | Gietz | 07-FEB-2012 | WGIEZT | 515.123.8181 | AC_ACCOUNT | 8300 |

Note that parent, **Department**, filters the results in **Employees**, the child.

Next, make the child, Employees, editable.

10. Make Employees report editable:
 - a. Access Page Designer. Click the **Edit Page XX** on the Developer Toolbar.

- b. In the Rendering tab, under **Employees**, select **Attributes**.
 - c. In the Property Editor, edit the following:
 - Edit, Enabled - select **Yes**.
 - Allowed Operations - Verify the following operations are enabled: **Add Row**, **Update Row**, and **Delete Row**.
11. Define the EMPLOYEE_ID column as the Primary Key Column:
- a. In the Rendering tab, under **Employees**, expand **Columns** and select **EMPLOYEE_ID**.
 - b. In the Property Editor, find **Source**, **Primary Key** and select **Yes**.
12. Click **Save**. To run the page, click **Save and Run Page**.

The child interactive grid, Employees, is now editable

The image shows two screenshots of an interactive grid interface. The top screenshot displays a grid with two columns: 'Department Name' and 'Manager Id'. The 'Accounting' row is selected, showing a 'Manager Id' of 205. The bottom screenshot displays a grid with columns: 'Employee Id', 'First Name', 'Last Name', 'Hire Date', 'Email', 'Phone Num...', and 'Job Id'. The row for employee 205 (Shelley Higgins) is selected, showing a 'Job Id' of AC_MGR.

Displaying Interactive Grids Conditionally

You can choose to have interactive grids display conditionally by editing region attributes.

To display an interactive grid conditionally:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, select the region.

The Property Editor displays the region attributes in the right pane.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

3. Edit Server-side Condition:
 - a. Find the **Server-side Condition** group.
 - b. For Type, select a condition type and follow the on-screen instructions.

When you edit attributes in Page Designer, the Messages tab displays errors and warnings you need to address. The Messages tab displays a red or yellow badge indicating the number of messages you need to address. Click the Messages tab to review the messages.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Managing Interactive Grid Attributes

Report Attributes control how an interactive grid works. Developers edit interactive grid Attributes to control if end users can edit the underlying data, configure report pagination, create error messages, configure the toolbar and download options, control if users can save public reports, and add Icon and Detail Views.

- [Editing Interactive Grid Attributes](#)
- [Making an Existing Interactive Grid Editable](#)
- [Controlling Interactive Grid Pagination](#)
- [Customizing the Interactive Grid Toolbar](#)
- [Enabling Users to Save Public Reports in Interactive Grids](#)
- [Restricting Who Can Save Public Reports](#)
- [Enabling Icon View in an Interactive Grid](#)
- [Enabling Detail View in an Interactive Grid](#)

 **See Also:**

["Editing Pages in Page Designer"](#)

Editing Interactive Grid Attributes

To edit interactive grid Attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Edit the appropriate attributes.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

See Also:

["Right Pane of Page Designer \(Property Editor\)"](#)

Making an Existing Interactive Grid Editable

End users can alter the report layout using Column menus and sort options, rearrange columns by dragging and dropping, and change how data displays using options on the Actions menu. However, developers determine whether the underlying data is read-only or editable by users. When an interactive grid is editable, end users can edit the underlying data, add rows, delete rows, and refresh rows. By default, editable interactive grids include a Save button on the toolbar. Developers can determine

To make an interactive grid editable:

1. View the page in Page Designer:

- a. On the Workspace home page, click the **App Builder** icon.
- b. Select an application.
- c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. In the Property Editor, find **Edit** and update the following attributes:
 - a. **Edit** - Select **Yes**.

Selecting **Yes** automatically creates a process to process the records. When Enabled is set to **Yes**, additional options appear.
 - b. **Allowed Operations** - Select supported data manipulation operations. Options include: **Add Row**, **Update Row**, **Delete Row**
 - c. **Allowed Row Operations Column** - Select a column where the column value indicates if the row can be updated and/or deleted.
 - d. **Lost Update Type** - Select how to protect data against lost updates.

Lost updates occur when two or more users try to update the same data, and the last update overwrites updates made by the previous user. To prevent this, updates can check the existing column value(s) in the database with the value(s) retrieved when the record was originally queried. For an example, see Page Designer Help.

 - **Row Values** - A checksum value is calculated for each row, when initially querying the data. The checksum is calculated by concatenating all of the updateable columns into a string and then generating a unique value. When committing the updated record, this checksum is compared to the checksum value.
 - **Row Version Column** - If your database table includes a column which is incremented every time a record is updated, preferably by a database trigger, then this column can be used instead of calculating checksums. Oracle does not recommend this option if your interactive grid updates data in multiple tables.
 - e. **Add Row If Empty** - Specify whether to add a new row if the interactive grid currently has no records returned. When enabled, the `No Data Found` message never displays since there will always be at least one record. If no updates are made to the record that was added it will not be inserted when the region is saved.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**See Also:**

"Right Pane of Page Designer (Property Editor)"

Controlling Interactive Grid Pagination

Pagination provides the end user with information about the number of rows displayed and the current position within the result set.

To change interactive grid pagination:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

4. Find **Pagination**.

Pagination attributes provide the end user with information about the number of rows displayed and the current position within the result set.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Under **Pagination**, edit the following attributes:
 - a. **Type** - Select an option:
 - **Page** - Rows display based on what the end users select in the Actions menu, **Format**, **Rows per Page** option. If there are additional rows, controls are added to the report footer which enables users to navigate forwards and backwards between row sets.

- **Scroll** - Rows display to fill the height of the interactive grid. As the end user scrolls, additional rows display as needed.
 - b. Show Total Row Count - Select **Yes** or **No**.
Selecting **Yes** displays the total row count in the report footer. Enabling this option requires an additional query which may hinder performance on very large data sets
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **Tip:**

Install the sample application, *Sample Interactive Grids*, to see interactive grid pagination examples. See "[Installing a Productivity and Sample App](#)."

Customizing the Interactive Grid Toolbar

All interactive grids include a toolbar at the top of the page. Developers can customize what controls display on the toolbar or remove it.

 **Note:**

Some of the options described in this procedure only apply if the interactive grid is editable. See "[Making an Existing Interactive Grid Editable](#)".

To customize the interactive grid toolbar:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

4. In the Property Editor, find the **Toolbar** group:
 - a. For Show, select **Yes** or **No** to determine if a toolbar displays above the report.

- b. For Buttons, select which buttons display in the toolbar.
Options include:
 - **Reset - Reset** removes any customizations (such as filters, column width, ordering, and so on) and reloads the report definition from the server.
 - **Save - Save** is only functional when an interactive grid is editable and the end user has authorization to add, update or delete. **Save** saves changes made to the interactive grid, without needing to save the whole page.
 - c. **Add Button Label** only displays when the interactive grid is editable. For **Add Button Label**, enter an alternative label for the Add button. The default label is **Add Row**.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.



Tip:

To view an example of adding a toolbar button, install the sample application, *Sample Interactive Grids*, and go to **Reporting, Add Toolbar Button**. See "[Installing a Productivity and Sample App](#)."

Enabling Users to Save Public Reports in Interactive Grids

Users can save an interactive grid as a Public report if the report developer enables that capability. Public reports display on the Saved Reports list on the toolbar and are available to all users. However, only the user who creates a Public report can save, rename, or delete it.



Tip:

The Save Public Report attribute is only available to authenticated users and can be further restricted by the defined Authorization scheme. See "[Restricting Who Can Save Public Reports](#)."

To enable end users to save Public reports:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
Page Designer appears.
2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.
The Property Editor displays the attributes. Attributes are organized in groups.

**Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. In the Property Editor:
 - a. Find **Enable Users To**.
 - b. Save Public Report - Select **Yes**
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Restricting Who Can Save Public Reports

Developers can select an authorization scheme to restrict who can save Public reports. To enable an end user to save a Public report, this authorization scheme must evaluate to TRUE for that user. If an authorization scheme is not selected, then any user may save Public reports.

To restrict who can save a Public report:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

**Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

4. In the Property Editor:
 - a. Find **Enable Users To**.
 - b. Save Public Report - Select **Yes**
 - c. Save Public Report Authorization - Select an authorization scheme.

Select an authorization scheme to restrict who can save public report definitions. To enable an end user to save a public report, this authorization scheme must evaluate to TRUE for that user. If you do not select an authorization scheme, then any user may save public reports.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

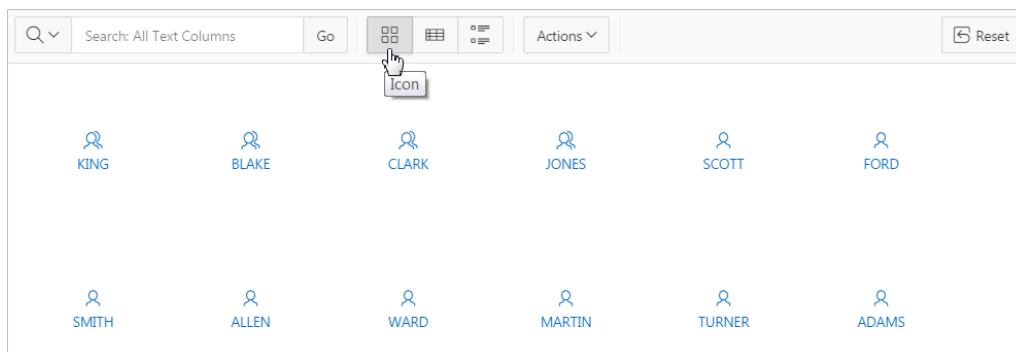
Enabling Icon View in an Interactive Grid

Interactive grids support different views of data. By default, interactive grids display data in a Grid view. Developers can optionally display data as icons. Enabling Icon view adds an Icon button to the select view button group.

- [About Icon View in an Interactive Grid](#)
- [Adding Icon View to an Interactive Grid](#)
- [Adding a Custom Icon View Link to an Interactive Grid](#)

About Icon View in an Interactive Grid

Interactive grids support different views of data. Users can switch between these views by clicking the buttons in the select view button group. The following illustration shows an interactive grid with Icon view enabled and the Icon button selected. To enable Icon view, you must identify the columns used to identify the icon, the label, and the target (that is, the link).



Viewing a Working Example of Icon View

To see a working example of this Icon view, install the sample application, *Sample Interactive Grids* and review **Reporting, Icon and Detail Views**.



See Also:

["Installing a Productivity and Sample App"](#)

Adding Icon View to an Interactive Grid

An important requirement of icon view is that each item in the view must have the same width and height and the item content must not overflow. The width and height are set with CSS rules on elements with a class of `a-IconList-item`.

To add an Icon view to an interactive grid:

To add an Icon view to an interactive grid:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.

- b. Select an application.
- c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

4. Find **Icon View**.
5. Configure the following **Icon View** attributes:

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- a. Show - Specify whether the Icon View is enabled. Select **Yes**.
 - b. Custom - Select **No**. Selecting **Yes** provides full control over the icon view item markup but also requires a full understanding of HTML markup.
 - c. Icon Type - Select the column type that holds the icon image source. Select **Icon CSS Classes**. You must also add a column to your SQL statement that has a value that is a CSS class for an icon (for example, a Font APEX icon class name). As an alternative, you can also select **Image in URL** and then specify the column that contains the URL and Icon Attributes.
 - d. Icon - Select the interactive grid column that returns the CSS classes for the icon image source.
 - e. Link Target - Click **Link Target** to display the Link Builder - Link Target dialog. The link target defined is followed when the user activates the icon link with mouse or keyboard.
 - f. Link Attributes - Enter HTML attributes that are associated with the display of each Link Target.
 - g. Label Column - Select the report column that returns the image label.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Adding a Custom Icon View Link to an Interactive Grid

A custom Icon View link provides greater flexibility in defining the content of the Icon View, including the ability to utilize more than one column.

To add custom Icon View link to an interactive grid:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

4. Find **Icon View**.
5. Configure the following **Icon View** attributes:

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- a. Show - Select **Yes**.
 - b. Custom - Select **Yes**. Selecting **Yes** provides full control over the icon view item markup but also requires a full understanding of HTML markup..
 - c. Icon Type - Select the column type that holds the icon image source. Select **Icon CSS Classes**. You must also add a column to your SQL statement that has a value that is a CSS class for an icon (for example, a Font APEX icon class name). As an alternative, you can also select **Image in URL** and then specify the column that contains the URL and Icon Attributes.
 - d. Icon - Select the interactive grid column that returns the CSS classes for the icon image source.
 - e. Link Target - Click **Link Target** to display the Link Builder - Link Target dialog. The link target defined is followed when the user activates the icon link with mouse or keyboard.
 - f. Link Attributes - Enter HTML attributes that are associated with the display of each Link Target.
 - g. Label Column - Select the report column that returns the image label.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Enabling Detail View in an Interactive Grid

Interactive grids support different views of data. By default, interactive grids display data in a Grid view. Developers can optionally display data as in a Detail view. Enabling Detail view adds an Detail button to the select view button group.

- [About Detail View in an Interactive Grid](#)
- [Adding a Detail View to an Interactive Grid](#)

About Detail View in an Interactive Grid

Interactive grids support different views of data. Users can switch between these views by clicking the buttons in the select view button group. The following illustration shows an interactive grid with Detail view enabled and the Detail button selected. Detail view displays column values using developer defined HTML markup as shown in the following illustration.

| Name | Manager | Job | Notes |
|-------|---------|-----------|---|
| KING | | PRESIDENT | |
| BLAKE | KING | MANAGER | Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. |
| CLARK | KING | MANAGER | |
| JONES | KING | MANAGER | |
| SCOTT | JONES | ANALYST | |
| FORD | JONES | ANALYST | |
| SMITH | FORD | CLERK | |
| ALLEN | BLAKE | SALESMAN | |
| WARD | BLAKE | SALESMAN | |

Viewing a Working Example of Icon View

To see a working example of this Icon view, install the sample application, *Sample Interactive Grids* and review **Reporting, Icon and Detail Views**.



See Also:

["Installing a Productivity and Sample App"](#)

Adding a Detail View to an Interactive Grid

To add a Detail view to an interactive grid:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Find **Detail View**.
5. Configure the following **Detail View** attributes:
 - a. Show - Select **Yes**.
 - b. Before Rows - Enter the HTML to be displayed before report rows.
 - c. For Each Row - Enter the HTML template to use for each report row in the Detail View. Use #COLUMN_NAME# substitution strings for column names and column labels.

The following substitution strings are available:

- &COLUMN_NAME. - substitution strings for column labels and names.
- &APEX\$ROW_ID. - the record identifier for the row.
- &APEX\$ROW_INDEX. - the record display sequence number.

Example:

```
<tr><td align="right">&ENAME_LABEL. :</td><td>&ENAME.</td></tr>  
<tr><td align="right">&JOB_LABEL. :</td><td>&JOB.</td></tr>
```

- d. After Rows - Enter the HTML to be displayed after report rows.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Managing Interactive Grid Column Attributes

Column attributes enable developers to control the display, features, and behavior of interactive grid columns. Developers edit Column attributes to alter nearly all aspects of column behavior, including altering the layout and appearance, creating validations, defining column links, creating column filters, and adding support for export and printing.

 **Note:**

Some settings are configured at runtime by the report developer and saved as part of the primary or alternate report (for example, column display order, the columns the report is sorted on, and column widths. To learn more about save options, see "[Saving Interactive Grids](#)."

- [Editing Common Grid Column Attributes](#)
- [Defining a Column as a List of Values for Filtering](#)
- [Creating a Column Link in an Interactive Grid](#)

 **See Also:**

["Editing Pages in Page Designer"](#)

Editing Common Grid Column Attributes

This section describes how to edit common grid columns attributes such as changing the column heading, determining how a column renders, how users can edit a column value, and how users can manipulate a column.

To edit interactive grid Column attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.
3. Select a column to edit.
4. Edit the appropriate attributes.

The Property Editor displays the attributes.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

5. To edit the column heading:
 - a. Find **Heading**.

- b. **Heading** - Enter the report column heading text.
 - c. **Alignment** - Select the heading alignment. Note that your alignment selection must be supported by the selected report template..
 - d. **Alternative Label** - Enter the alternative label to use in dialogs and in the Single Row View. Use an alternative label when the heading contains extra formatting, such as HTML tags, which do not display properly.
6. To control how column values display and for editable grids how the column value is edited:
- a. Find **Identification**.
 - b. **Type** - Select how the column is rendered.

Different settings are available depending on the selected Type. For example, for an editable grid, select the Type **Display Only** to make a column uneditable. Choose the Type **Hidden** for primary key columns that should not be seen or other columns that are only used in templates such as icon or detail view, link targets, or columns of type HTML Expression.

 **Note:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. To control how users can manipulate a column:
- a. Find **Enable Users To**.
 - b. **Sort/Control Break/Aggregate** - Select whether end users can use this column in sorts, control breaks, and aggregates. Select **Yes** or **No**.

 **Tip:**

Enabling these operations can cause performance issues, especially on large data sets or where appropriate column indexes are not available.

8. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

["Right Pane of Page Designer \(Property Editor\)"](#)

Defining a Column as a List of Values for Filtering

You can define an interactive grid column as a list of values to improve the speed of built-in filter tools.

To define a report column as a list of values:

1. View the page in Page Designer:

- a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.
3. Select a column to edit.
The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. In the Property Editor, find **Column Filter**. Specify the following:
 - a. Enabled - Specify whether end users can filter this column. Select **Yes** or **No**.
 - b. LOV Type - Select how to derive the list of values for the filter when a user clicks on the column header or in filter dialogs. Options include:
 - **None** - Disable any values being displayed in the filter. This option is appropriate on large columns, generally with most values being over 50 characters, or when a list would not be very usable for selecting a specific record. Instead the end user can just type in a partial value and then view the results.
 - **Distinct Column** - Distinct column values are displayed for VARCHAR2 and NUMBER columns. DATE and TIMESTAMP columns utilize the built-in predefined date ranges.
 - **SQL Query** - Values are derived from the SQL query definition.
 - Shared Component - Values are derived from an existing list of values.
 - **Static Values** - Values are derived from a static list of values.
 - **PL/SQL Function Body returning SQL Query**- Values are derived from the PL/SQL function body that returns a SQL Query.
 - **Use List of Values** - Values are derived from list of values.
6. To save your changes click **Save**.
7. To save and run the page, click **Save and Run Page**.

Creating a Column Link in an Interactive Grid

Use Column Attributes to create a link from a report to another page in your application or to a URL.

To create a column link to another page:

1. View the page in Page Designer:

- a. On the Workspace home page, click the **App Builder** icon.
- b. Select an application.
- c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.
3. Select the desired column.

The Column attributes appear in the Property Editor. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

4. Find **Identification**. From Type, select **Link**.
5. Find **Link** and click **No Link Defined**.
The Link Builder - Target dialog appears. The Target Type you select determines the steps that follow.
6. To create a link to another page, in the Link Builder - Target dialog:
 - a. Type - Select **Page in this application**.
 - b. Page - Specify the target page number.
 - c. Set Items - Select a Name and Value to specify session state for an item.
 - d. Clear Session State, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple pages, enter a comma-delimited list of page numbers.
 - e. Rest Pagination - Select **Yes** to reset pagination for this page.
 - f. Advanced, Request - Specify the request to be used.
 - g. Click **OK**.
7. To create a link to another page in another application, in the Link Builder - Target dialog:
 - a. Type - Select **Page in a different application**.
 - b. Application - Specify the application.
 - c. Page - Specify the target page number.
 - d. Set Items - Select a Name and Value to specify session state for an item.
 - e. Clear Session State, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple pages, enter a comma-delimited list of page numbers.
 - f. Rest Pagination - Select **Yes** to reset pagination for this page.
 - g. Advanced, Request - Specify the request to be used.
 - h. Click **OK**.

8. To create a link to a URL, in the Link Builder - Target dialog:
 - a. Type - Select **URL**.
 - b. URL - Enter the URL address.
 - c. Click **OK**.
9. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

About Customizing Interactive Grids in a Running Application

When running an application, users can alter the report layout using Column menus and sort options, rearrange columns by dragging and dropping, and change how data displays using options on the Actions menu.

When an interactive grid is editable, users can also edit the underlying data, add rows, delete rows, and refresh rows. After users customize an interactive grid they can save it. However, the available save capabilities differ depending upon the user type.

See Also:

- ["Making an Existing Interactive Grid Editable"](#)
- ["Saving Interactive Grids"](#)
- *Oracle Application Express End User's Guide*

Saving Interactive Grids

After users customize an interactive grid they can save it as either a Private or Public report.

Only the user who creates a **Private** report can view, save, rename, or delete it. Report developers have additional save capabilities in that they can save the **Primary** report (that is, the report that displays to all users), or create an **Alternative** report to test multiple report layouts.

Tip:

To see a working examples of saved interactive grids, install the sample application, *Sample Interactive Grids* and review *Reporting, Saved Reports* . See ["Installing a Productivity and Sample App."](#)

- [How the User Type Effects Available Save Options](#)
- [About Configuration Dependencies When Saving Interactive Grids](#)
- [Saving a Private Interactive Grid](#)
- [Saving a Public Interactive Grid](#)
- [Saving or Renaming a Primary Interactive Grid](#)

- [Saving an Alternative Interactive Grid](#)
- [Renaming a Saved Interactive Grid Report](#)
- [Deleting a Saved Interactive Grid Report](#)
- [Resetting a Saved Interactive Grid Report](#)

How the User Type Effects Available Save Options

After users customize an interactive grid they can save it. However, the available save capabilities differ depending upon the user type.

End Users

End users can save an interactive grid as follows:

- **Private** report. Only the user who creates a Private report can view, save, rename, or delete it. All users (that is, end users and developers) can create interactive grid Private reports.
- **Public** report. All users can view a Public report. However, only the user who creates a Public report can save, rename, or delete it. All other users can view a Public report and save it under a new report name.



Tip:

End users can only save Public reports if the report developer has enabled that capability. See "[Enabling Users to Save Public Reports in Interactive Grids](#)."

Developers

In addition to Private and Public reports, a developer can save two other report formats:

- **Primary** report. The Primary report displays to all users. Only a developer can save a new Primary report, rename it, or delete it.
- **Alternative** report. An Alternative report enables developers to create multiple report layouts. Only developers can save, rename, or delete an Alternative report.

About Configuration Dependencies When Saving Interactive Grids

To save a customized version of an interactive grid, users select the Actions menu and then click **Report** and then **Save** or **Save As**. End users can only access Save options if the application developer defines an authentication scheme at the application-level.

Other configuration issues that impact a user's ability to save an interactive grid include:

- **Interactive Grid Toolbar Availability.** Developers can edit the interactive grid Attribute, **Toolbar** to determine if an interactive grid includes a toolbar. If the toolbar is disabled, all the options on the Actions menu, including Save and Save As, are unavailable.

- **Support for Saving Public Reports.** By default, end users cannot save Public reports. To enable support for Public reports, developers edit the report Attribute, **Enables Users To, Save Public Report**.

 **See Also:**

- ["Establishing User Identity Through Authentication"](#)
- ["Customizing the Interactive Grid Toolbar"](#)
- ["Enabling Users to Save Public Reports in Interactive Grids"](#)

Saving a Private Interactive Grid

All users (that is, both end users and developers) can save a Private interactive grid. Only the user who creates a Private interactive grid can view, save, rename, or delete it.

To save a Private interactive grid:

1. Run the interactive grid.
2. Customize the report (for example, show, hide, or move columns, create filters, and so on).
3. Click the **Actions** menu.
4. Select **Report** and then **Save As**.
5. In the Report - Save As dialog:
 - Type - Select **Private**.
 - Name - Enter a name for the report.
 - Click **Save**.

Private reports display on the Saved Reports list on the toolbar under the heading, Private.

Saving a Public Interactive Grid

All users (that is, both end users and developers) can save a Public interactive grid. However, only the user who creates a Public interactive grid can save, rename, or delete it. Users who did not create the public report can view it and also have the option to make their own copy of it using the Save As option.

 **Note:**

End users can only save Public interactive grids if the report developer has enabled that capability. See ["Enabling Users to Save Public Reports in Interactive Grids"](#).

To save a Public interactive grid:

1. Run the interactive grid.
2. Customize the report (for example, show, hide, or move columns, create filters, and so on).
3. Click the **Actions** menu.
4. Select **Report** and then **Save As**.
5. In the Report - Save As dialog:
 - Type - Select **Public**.
 - Name - Enter a name for the report.
 - Click **Save**.

Public reports display on the Saved Reports list on the toolbar under the heading, Public.

Saving or Renaming a Primary Interactive Grid

A Primary interactive grid displays to all users. Only developers can save or rename Primary reports. It is impossible to delete a primary report.

To save or rename a Primary interactive grid:

1. Run the interactive grid as a developer.
2. To customize and then save a primary interactive grid:
 - a. Customize the report (for example, show, hide, or move columns, create filters, and so on).
 - b. Click the **Actions** menu.
 - c. Select **Report** and then **Save**.
3. To rename a primary report:
 - a. Click the **Actions** menu.
 - b. Select **Report** and then **Edit**.
The Report - Edit dialog appears.
 - c. In Name, enter a new name and click **Save**.
4. Select **Report** and then **Save**.

If multiple reports have been redefined, the Primary report displays on the Saved Reports list on the toolbar under the heading, Default.

Saving an Alternative Interactive Grid

Alternative reports enable a developers to optionally define multiple report layouts, which are available to all users (subject to any authorization scheme, if defined). Only developers can save, rename or delete an alternative report. Only the developer who creates an interactive grid can save, rename, or delete an Alternative report.

To save an Alternative interactive grid:

1. Run the interactive grid as a developer.
2. Customize the report (for example, show, hide, or move columns, create filters, and so on).

3. Click the **Actions** menu.
4. Select **Report** and then **Save As**.
 - Type - Select **Alternative**.
 - Name - Enter a name for the report.
 - Authorization - Select an alternative authorization scheme to restrict who can view this report.
 - Click **Save**.

An Alternative report displays on the Saved Reports list on the toolbar under the heading, Default.

Renaming a Saved Interactive Grid Report

All users can rename the interactive grid reports they create.

To rename an interactive grid report:

1. Run the interactive grid.
2. From the Saved Report list on the toolbar, select the report.
3. Click the **Actions** menu and then select **Report** and then **Edit**.
4. In the Report - Edit dialog:
 - a. Type - Select the type of report to be edited.
 - b. Name - Enter a new name.
 - c. Click **Save**.

The new report name displays in the Saved Reports list on the toolbar.

Deleting a Saved Interactive Grid Report

All users can delete the interactive grid reports they create.

To delete an interactive grid report:

1. Run the interactive grid.
2. From the Saved Report list on the toolbar, select the report.
3. Click the **Actions** menu and then select **Report** and then **Delete**.

A confirmation dialog appears.
4. Confirm your request and click **OK**.

The report disappears from the Saved Reports list on the toolbar.

Resetting a Saved Interactive Grid Report

All users can reset an interactive grid report to the last saved version.

To reset an interactive grid report:

1. Run the interactive grid.
2. From the Saved Report list on the toolbar, select the report.

3. Customize the report (for example, show, hide, or move columns, create filters, and so on).
4. Reset the report to the last saved version. Click the **Actions** menu and then select **Report** and then **Reset**.

 **Tip:**

Alternatively, you can also click the **Reset** button on the toolbar.

The revised report appears.

Managing Interactive Reports

An interactive report is the formatted result of a SQL query that is only supported on Desktop pages. End users can customize the report layout and data displayed by selecting options on the Actions menu. Developers control how an interactive report works by editing the following attributes in Page Designer: region attributes, report Attributes, and Column attributes.

 **Tip:**

Interactive grids include the customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse and keyboard. To learn more see "[Interactive Grid](#)".


- [Managing Interactive Report Region Attributes](#)
Each application page contains one or more regions. A region is an area on a page that serves as a container for content. Developers edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.
- [Managing Interactive Report Attributes](#)
Report Attributes control how an interactive report works. Developers edit interactive grid Attributes to control if end users can edit the underlying data, configure report pagination, create error messages, configure the toolbar and download options, control if users can save public reports, and add Icon and Detail Views.
- [Managing Interactive Report Column Attributes](#)
Column attributes enable developers to control the display, features, and behavior of interactive grid columns. Developers edit Column attributes to alter nearly all aspects of column behavior, including altering the layout and appearance, creating validations, defining column links, creating column filters, and adding support for export and printing.
- [Customizing Interactive Reports in a Running Application](#)
When running an application, users (that is, both end users and developers) can alter the layout of interactive reports data by choosing the columns they are interested in, applying filters, highlighting, and sorting. They can also define breaks, aggregations, charts, group bys, and add their own computations.

- [Linking to Interactive Reports](#)
Developers can use the syntax in the `Request` value and `ItemNames` section of the URL to link to interactive reports.

Managing Interactive Report Region Attributes

Each application page contains one or more regions. A region is an area on a page that serves as a container for content. Developers edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.

- [Editing Interactive Report Region Attributes](#)
- [Displaying Interactive Reports Conditionally](#)

 **See Also:**
"Editing Pages in Page Designer"

Editing Interactive Report Region Attributes

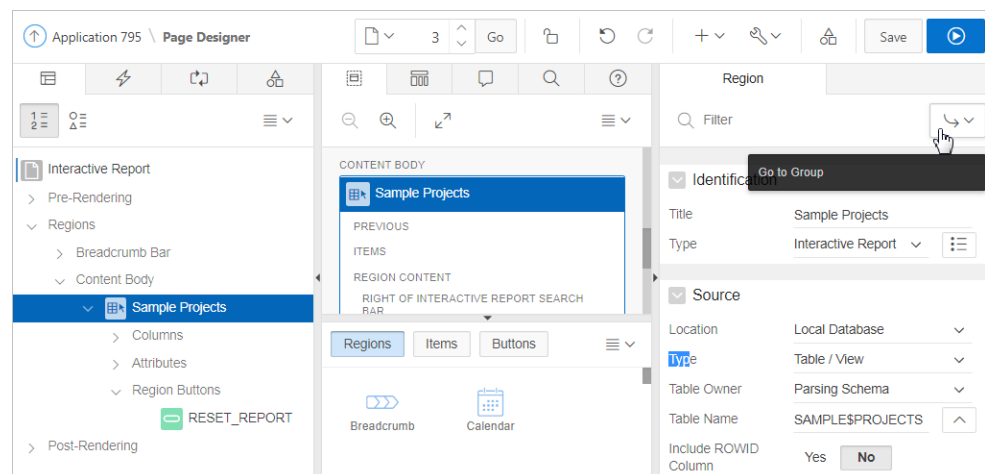
To edit region attributes in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, select the region.

The Property Editor displays the region attributes in the right pane.



3. To find a group or attribute:

- **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

["Right Pane of Page Designer \(Property Editor\)"](#)

Displaying Interactive Reports Conditionally

You can choose to have interactive reports display conditionally by editing region attributes.

To display an interactive report conditionally:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate and select the region containing the report.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group..

3. Configure Server-side Condition:
 - a. Find **Server-side Condition**.
 - b. For Type, select a condition type and follow the on-screen instructions

Page Designer notifies you of errors and warnings you need to address before you can save the page. The Messages tab displays a red or yellow badge indicating the number of messages you need to address

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **Tip:**

Asynchronous JavaScript and XML (Ajax) is used throughout interactive reports. Because Ajax is asynchronous, the value evaluated for conditional display must be in the session. To display an interactive report conditionally based on specified value, use a page item instead of using a `:REQUEST` value. Using `:REQUEST` in a conditional display causes the interactive report function incorrectly.

Managing Interactive Report Attributes

Report Attributes control how an interactive report works. Developers edit interactive grid Attributes to control if end users can edit the underlying data, configure report pagination, create error messages, configure the toolbar and download options, control if users can save public reports, and add Icon and Detail Views.

- [Editing Interactive Report Attributes](#)
- [Managing Link Columns](#)
- [Customizing the Interactive Report Search Bar](#)
- [Customizing the Actions Menu](#)
- [Configuring Actions Menu Download Options](#)
- [Controlling Interactive Report Pagination](#)
- [Enabling Icon View](#)
- [Enabling Detail View](#)
- [Configuring Advanced Attributes for Interactive Reports](#)

 **See Also:**

["Editing Pages in Page Designer"](#)

Editing Interactive Report Attributes

To edit interactive report Attributes in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Edit the appropriate attributes.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

["Right Pane of Page Designer \(Property Editor\)"](#)

Managing Link Columns

A Link Column displays on the left side of an interactive report. Developers can configure a Link Column to link to a single row view, a custom target, or be excluded from the report.

 **Tip:**

A Link Column cannot be sorted, hidden, or moved by an end user and does not interact with the standard column link defined on the Column Attributes page.

- [Linking to a Single Row View](#)
- [Linking to a Specific Page](#)
- [Linking to a URL](#)
- [Excluding a Link Column](#)

Linking to a Single Row View

To link to a single row view in an interactive report:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under Region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Configure Link attributes:
 - a. Find the **Link** group.
 - b. Link Column - Select **Link to Single Row View**.
 - c. Uniquely Identify Rows by - Select **ROWID** or **Unique Column**.
If you select, **Unique Column**, specify the column in the Unique Column field.
 - d. Link Icon - Accept the default or specify the path to another icon.
 - e. Link Attributes - Optionally specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
 - f. Authorization Scheme - Select an authorization scheme to associate with the Link Column. This authorization scheme must evaluate to TRUE in order for the Link Column to be rendered in the report.
 - g. Condition Type - Select a condition type from the list that must be met in order for this link to be rendered.
5. Find the **Single Row View** group:
 - a. Exclude Null Values - Specify whether null columns are hidden on the Single Record View
 - b. Only Displayed Columns - Specify whether only those columns currently displayed are displayed in the Single Record View. If you specify No then hidden report columns may also be shown on the Single Record View.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **Tip:**

If an interactive report links to single row view and the link icon is missing ALT text or has empty ALT text, the `alt="Single Row View"` gets included automatically to improve accessibility. This can be further improved by making the ALT text something specific to the row, for example `alt="View Row: #COLUMN_NAME#" (where COLUMN_NAME should be a column that helps uniquely identify the current row).`

Linking to a Specific Page

To link to a specific page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under Region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Configure Link attributes:
 - a. Find the **Link** group.
 - b. Link Column - Select **Link to Custom Target**.
 - c. Target - Click **No Link Defined**.

The Link Builder - Target dialog appears.

- Target Type - Select **Page in this Application**.
- Page - Select the target page number.
- Set Items - Select a **Name** and **Value** to specify session state for an item.
- Clear Session State, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple page, enter a comma-delimited list of page numbers.
- Clear Session State, Reset Pagination - To reset the pagination for this page, select **Yes** for Reset Pagination.

- Advanced, Request - Specify the request to be used.
 - Click **OK**.
- d. Link Icon - Enter the HTML to be used for the link. Use an image tag to display images, or pick from the list of default images. If not defined, #IMAGE_PREFIX#ws/small_page.gif is used.
 - e. Link Attributes (Optional) - Enter additional column link attributes to be included in the tag (for example, a link target, classes, or styles).

To view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Linking to a URL

To link to a URL:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under Region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Configure Link attributes:
 - a. Find the **Link** group.
 - b. Link Column - Select **Link to Custom Target**.
 - c. Target - Click the **No Link Defined**.

The Link Builder - Target dialog appears.
 - Target Type - Select **URL**.
 - Click **OK**.
 - d. Link Icon - Enter the HTML to be used for the link. Use an image tag to display images, or pick from the list of default images. If not defined, #IMAGE_PREFIX#ws/small_page.gif is used.
 - e. Link Attributes (Optional) - Enter additional column link attributes to be included in the tag (for example, a link target, classes, or styles).

To view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Excluding a Link Column

To exclude a column link:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under Region, select the **Attributes** node.

The Property Editor displays the attribute. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Configure Link attributes:
 - a. Find the **Link** group.
 - b. Link Column - Select **Exclude Link Column**.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Customizing the Interactive Report Search Bar

All interactive reports include a search bar at the top of the page. Developers can customize what controls display or remove the search bar.

- [Interactive Report Search Bar](#)
- [Customizing the Search Bar](#)
- [Removing the Interactive Report Search Bar](#)

Interactive Report Search Bar

All interactive reports include a search bar at the top of the page which includes the following controls: a Select columns to search icon, Text area, Go button, and Actions menu.

| <input type="text"/> Go 1. Primary Report ⌵ ⌵ ⌵ Actions ⌵ Create | | | | |
|--|--|-------------|----------|------------|
| Name | Description | Status | End Date | Start Date |
| Configure Web Development Tool Environment | Determine the hardware and software required to develop with Web development tool. | COMPLETED | 30-JUL | 26-AUG |
| Train Developers on Web development tool | Ensure all developers who will be developing with the new tool get the appropriate training. | COMPLETED | 14-AUG | 26-AUG |
| Migrate Legacy Applications | Move the data and redevelop the applications currently running on top of legacy servers | IN-PROGRESS | - | 26-AUG |
| Develop Partner Portal POC | Develop a proof of concept that partners can use to work more collaboratively with us. | IN-PROGRESS | - | 26-AUG |
| Develop Production Partner Portal | Develop the production app that partners can use to work more collaboratively with us. | ASSIGNED | - | 26-AUG |
| | | | | 1 - 5 |

If users have saved multiple reports, a saved report select list displays to the right of the **Go** button. Users can use this list to select a specific saved report. If multiple views are configured, a view selection button group also displays which enables the end user to switch views. You can remove or customize the search bar in Page Designer by configuring report attributes in the Property Editor.

See Also:

["Customizing the Search Bar"](#) and ["Removing the Interactive Report Search Bar"](#)

Customizing the Search Bar

To customize the search bar:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

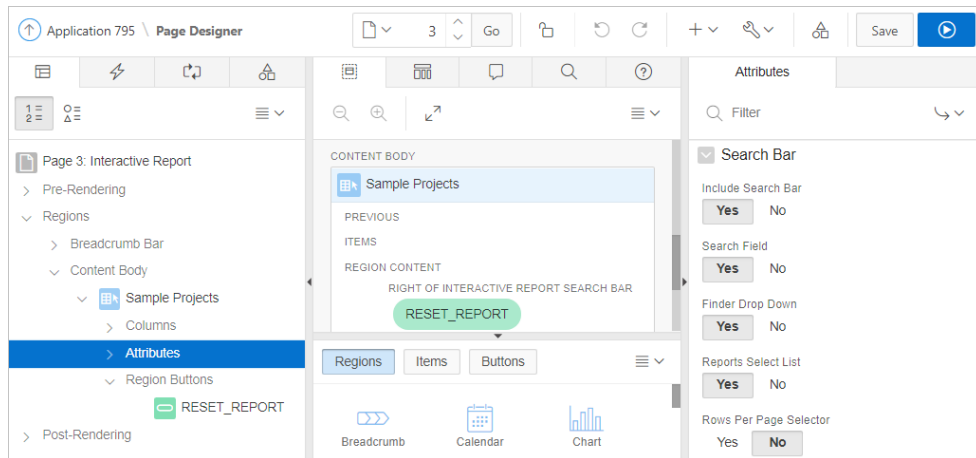
Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the report attributes for the page. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Find Search Bar.**5. Under Search Bar, select the options to be included by selecting **Yes** or **No**:**

- **Include Search Bar** - Select **Yes** to include a search bar above the report. If you include the search bar, you can also fully customize which functions to display. Select **No** to not include the Search bar.
- **Search Field** - Specify whether the search field is included within the search bar.
- **Finder Drop Down** - Specify whether the Finder drop down is included at the beginning of the Search field within the search bar. The Finder drop down is a list of columns that enables users to limit their search to a specific column. If a column is selected then the search is only performed against the values in the selected column. Within a running interactive report, this option is identified as **Select columns to search**.
- **Reports Select List** - Specify whether the Reports select list displays in the search bar. The Reports select list only displays if a report, other than the primary report, is accessible to the user.
- **Rows Per Page Selector** - Specify whether a separate drop down row selector is included within the search bar. Do not select both this attribute and **Rows Per Page**.
- **Search Button Label** - Enter the text for the search button label.
- **Maximum Rows Per Page** - Enter the maximum number to display in the Rows Per Page selector.

6. To save your changes click **Save. To save and run the page, click **Save and Run Page**.**

**See Also:**

["Saving Interactive Reports"](#)

Removing the Interactive Report Search Bar

To remove the search bar from in interactive report:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the report attributes for the page. Attributes are organized in groups.

**Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Find **Search Bar**.
5. For **Include Search Bar**, select **No**.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Customizing the Actions Menu

The Actions menu appears to the right of the Go button on the Search bar in an interactive report. Users use the Actions menu to customize how report information displays. Developers control what options display on Action menu in an interactive a report.

**Tip:**

The structure of the interactive report Actions menu changed in release 5.1 to be consistent with interactive grids. To alter interactive report Actions menu to use the previous (or legacy) structure, go to **Shared Components, Component Settings**, and configure the **Interactive Report** component. See ["Configuring Interactive Report Action Menu Structure"](#).

To customize the Actions menu:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the report attributes for the page. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Configure the options under Actions Menu:
 - a. Find **Actions Menu**.
 - b. Configure the options by selecting **Yes** or **No**.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Configuring Actions Menu Download Options

The Download option on the Actions menu enables users to download an interactive report as a comma-delimited file (CSV) format, HTML, Microsoft Excel (XLS) format, Adobe Portable Document Format (PDF), Microsoft Word Rich Text Format (RTF), or as HTML attached to an email. Microsoft Excel (XLS), Microsoft Word Rich Text Format (RTF) and Adobe Portable Document Format (PDF) formats require the configuration of a print server. Microsoft Excel and Microsoft Word formats also require BI Publisher.

 **WARNING:**

There is a 32K limit on downloads from the reports to advanced formats, such as PDF, XLS, and Word. If the end user has too many columns displayed then they may encounter an error when attempting to download the data to these advanced formats. Downloads may also take considerable time to complete for large data sets.

 **Note:**

The Group By and Pivot view do not support download formats of XLS, PDF, RTF even if they are enabled and report server is configured.

- [Enabling Download Formats](#)
- [Enabling CSV Download](#)
- [Enabling Email Download](#)

Enabling Download Formats

To configure download formats:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Edit the following attributes:
 - a. Search Bar, Include Search Bar - Select **Yes**.
 - b. Actions Menu, Download - Select **Yes**.

 **Tip:**

Selecting **No** for Download prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all Download Formats described in the next step.

- c. Download, Download Formats - Select download formats:
 - **CSV**
 - **HTML**
 - **Email**

- **XLS.**
- **PDF**
- **RTF**

 **Tip:**

The Download option only appears on the Actions menu if a file format is selected. If a report server has not been defined within Instance Administration then certain options do not display. To learn more, see "Configuring Report Printing" *Oracle Application Express Administration Guide*.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

["About Printing Reports to PDF"](#)

Enabling CSV Download

To enable support for CSV download, you must first enable **Automatic CSV Encoding** on the Edit Globalization Attributes page. See "[Accessing the Globalization Attributes Page](#)" and "[Automatic CSV Encoding](#)."

To enable the Enable CSV download in an interactive report:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. **Search Bar, Include Search Bar** - Select **Yes**.

5. **Actions Menu, Download** - Select **Yes**.

 **Tip:**

Selecting **No** for Download prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all Download Formats described in the next step.

6. **Download, Download Formats:**

a. **Download Formats** - Select **CSV**.

 **Tip:**

The Download option only appears on the Actions menu if a file format is selected.

b. **CSV Separator** - Define the column separator. If no value is entered, a comma or semicolon is used depending on your current NLS settings.

c. **CSV Enclosed By** - Enter a delimiter character. This character delineates the starting and ending boundary of a data value. Default delimiter is double quotation marks (" ").

d. **Filename** - Enter a name for the downloaded document. Do not specify a file extension, the extension is added automatically. If you leave this blank, the region name is used as the file name.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Enabling Email Download

You can configure the Download option to send a static HTML report to specific email addresses.

 **Tip:**

Email download is only supported on authenticated pages. If you enable Email download on a public page, the Email download option does not display in Download dialog.

To enable support for email download, an Oracle Application Express administrator must configure email at the Instance level. See "Configuring Mail" in *Oracle Application Express Administration Guide*.

To enable email download:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. **Search Bar, Include Search Bar** - Select **Yes**.
6. **Actions Menu, Download** - Select **Yes**.

 **Tip:**

Selecting **No** for Download prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all Download Formats described in the next step.

7. **Download Formats** - Select **Email**.

 **Tip:**

The Download option only appears on the Actions menu if a file format is selected.

8. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **Note:**

Email sent from an Email Download contains a system generated email signature identifying who sent the email. This signature cannot be removed.

**See Also:**

["About Emailing from an Interactive Report"](#)

Controlling Interactive Report Pagination

Pagination provides the end user with information about the number of rows and the current position within the result set. You control how pagination displays by making selections from Pagination attributes on the Attributes page in the Property Editor.

To edit interactive report pagination:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

**Tip:**

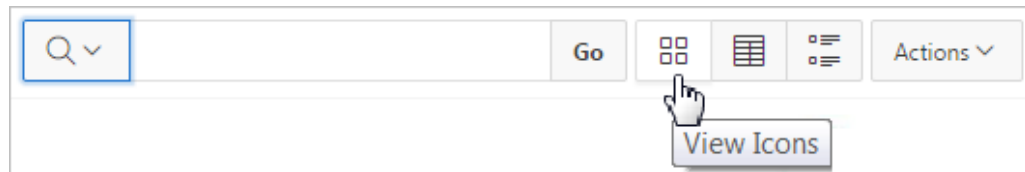
To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Configure Pagination attributes:
 - a. Find **Pagination**.
 - b. Type - Select the pagination type for this report. Often only a certain number of rows of a report display on a page. In order to include additional rows, the application end user must to navigate to the next page of the report.
 - c. Display Position - Select where the report pagination displays.

Pagination can be shown above or below the report (or both) and can be positioned on the left or right.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Enabling Icon View

By default, most interactive reports display as a report. You can optionally display columns as icons. When configured to display columns as icons, a View Icons icon displays on the Search bar.



To use Icon view, you must identify the columns used to identify the icon, the label, and the target (that is, the link). As a best practice, set these columns to display as hidden since they are typically not useful for end users.

- [Adding a View Icon Button](#)
- [Adding a Custom Icon View Link](#)

Adding a View Icon Button

To add Icon view to an interactive report:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the report attributes for the page. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Configure Icon View.
 - a. Find **Icon View**.
 - b. Configure the Icon View attributes:

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- i. Show - Select **Yes** adds a View Icons icon to the Search bar.
- ii. Custom Icon View - Specify whether you wish to define a custom link for the Icon View. This option provides greater flexibility in defining the

content of the Icon View, including the ability to utilize more than one column. Select **Yes** or **No**.

- iii. Columns Per Row - Select the number of icons to display per row in Icon View. A value of 5 displays 5 icons per row of the report.
 - iv. Link Column - Select the column that returns the link target of the icon.
 - v. Image Source Column - Identify the column that returns the image source.
 - vi. Label Column - Select the column that returns the image label.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.



See Also:

["Adding a Custom Icon View Link"](#)

Adding a Custom Icon View Link

A custom Icon View link provides greater flexibility in defining the content of the Icon View, including the ability to utilize more than one column.

To add a custom Icon View link:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the report attributes for the page. Attributes are organized in groups.



Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Find **Icon View**.
5. Configure Icon View attributes:
 - a. Show - Select **Yes**.
 - b. Custom - Select **Yes**.
 - c. Columns Per Page - Select the number of icons to display per row in Icon View. A value of 5 displays 5 icons per row of the report.

- d. Custom Link - Enter the column link for the Icon view. Use substitution strings for column names, for example:

```
<a href="#EMP_LINK#"><br>#ENAME#</a>
```

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

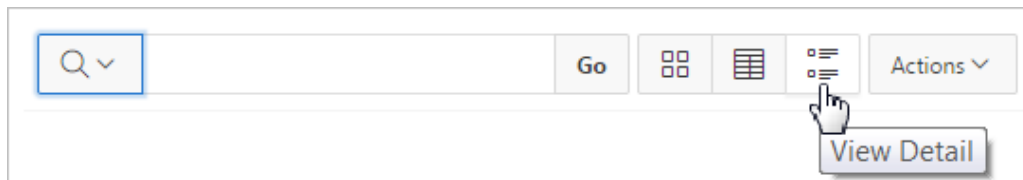


See Also:

["Adding a Custom Icon View Link"](#)

Enabling Detail View

By default, most interactive reports display as a report. You can optionally display details about each column. When configured, a View Details icon displays on the Search bar.



To add Details view to an interactive report:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.



Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Configure Detail View attributes:
 - a. Find **Detail View**.

Enter an alphanumeric alias for this interactive report region. The alias must be unique within the application. You can use an alias to reference an interactive report region in your custom API call. For example, you can write a function to return the `INTERACTIVE_REPORT_ID` by querying the `ALIAS` column in the `APEX_APPLICATION_PAGE_IR` view. Since the `INTERACTIVE_REPORT_ID` value can change when you export or import the application to different instance, referencing interactive report regions using an alias may be preferable.

b. Report ID Item:

Select a page or application item whose value stores the saved report ID. Interactive report uses this item value to link to a saved report. For example, you can define list of saved reports with links with defined item name and saved report ID as its value to link to a saved default reports, public or private reports. The saved report IDs can be retrieved from the `APEX_APPLICATION_PAGE_IR_RPT` view.

c. Email From Address:

Enter a valid email address to use as the From address when sending email from this report. Emails are sent from this report for Downloads of type Email and for Subscriptions. The value can be a literal string containing a valid email address or a static substitution reference defined in the application using substitution syntax.

Oracle does not recommend using an item substitution at the application or page-level since it only works in an email download, but not for subscriptions.

Examples:

```
john.doe@abc.com  
&MY_APP_EMAIL_FROM.  
&APP_EMAIL.
```

- 5.** To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

About Emailing from an Interactive Report

You can define the from address for emails sent from an interactive report in two ways:

- Application-level — Edit the **Application Email From Address** attribute on the Edit Application Definition page.
- Report-level — Access the report Attributes. Under **Advanced**, edit the **Email from Address** attribute.

In either case, the value can be a literal string containing a valid email or an application substitution defined at the application-level using substitution syntax or `&APP_EMAIL.` substitution which returns the Application Email From Address defined at the application-level. Examples:

```
john.doe@abc.com  
&MY_APP_EMAIL_FROM.  
&APP_EMAIL.
```

 **Note:**

Oracle does not recommend using an item substitution at the application or page-level since it only works in email download, but not for subscriptions.

 **See Also:**

- ["Accessing the Edit Application Definition Page"](#)
- ["Configuring Advanced Attributes for Interactive Reports"](#)
- ["Enabling Email Download"](#)

Managing Interactive Report Column Attributes

Column attributes enable developers to control the display, features, and behavior of interactive grid columns. Developers edit Column attributes to alter nearly all aspects of column behavior, including altering the layout and appearance, creating validations, defining column links, creating column filters, and adding support for export and printing.

 **Note:**

Some settings are configured at runtime by the report developer and saved as part of the primary or alternate report (for example, column display order, the columns the report is sorted on, and column widths. To learn more about save options, see ["Saving Interactive Grids."](#)

- [Editing Interactive Report Column Attributes](#)
- [Creating a Column Link in an Interactive Report](#)
- [Defining a Column as a List of Values in an Interactive Report](#)
- [Formatting Columns in an Interactive Report](#)
- [About Filtering on Conditional Links in Interactive Report Columns](#)

 **See Also:**

["Editing Pages in Page Designer"](#)

Editing Interactive Report Column Attributes

Edit Column attributes to precisely control the report layout. For example, you can use these attributes to alter a column heading, change column positioning, hide a column, or control how users can manipulate a column.

To edit column attributes:

To edit column attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.

3. Select a column to edit.

The Property Editor displays the attributes for that column. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Edit the appropriate attributes.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To edit the column heading, find the **Heading** group:
 - a. Heading - Enter the report column heading text.
 - b. Alignment - Select the heading alignment. Note that your alignment selection must be supported by the selected report template.
6. To control how users can manipulate a column:
 - a. Find the **Enable Users To** group.
 - b. For Enable Users To - For each attribute select **Yes** or **No** to control how users can manipulate the report.
7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

See Also:

["About Filtering on Conditional Links in Interactive Report Columns"](#) and ["Right Pane of Page Designer \(Property Editor\)"](#)

Creating a Column Link in an Interactive Report

Use Column Attributes to create a link from a report to another page in your application or to a URL.

To create a column link to another page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.

The columns appear in the Rendering tab.

3. Select a column to edit.

The Property Editor displays the attributes for that column. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Find **Identification**. From Type, select **Link**

5. Find **Link** and click **No Link Defined**.

The Link Builder - Target dialog appears.

6. To create a link to another page, in the Link Builder - Target dialog:

- a. Type - Select **Page in this Application**.
- b. Page - Specify the target page number.
- c. Set Items - Select a Name and Value to specify session state for an item.
- d. Clear Session State, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple page, enter a comma-delimited list of page numbers.
- e. Rest Pagination - Select **Yes** to reset pagination for this page.v
- f. Advanced, Request - Specify the request to be used.
- g. Click **OK**.

7. To create a link to a URL, in the Link Builder - Target dialog:

- a. Type - Select **URL**.
- b. URL - Enter the URL address.
- c. Click **OK**.

8. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Defining a Column as a List of Values in an Interactive Report

Add a list of values in an interactive report to improve the speed of built-in filter tools.

To define a report column as a list of values:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region containing the report.
3. In the Rendering tab, locate the region and expand the Columns node.

The columns appear in the Rendering tab.

4. Select a column to edit.

The Property Editor displays the attributes for that column. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Configure Column Filter:
 - a. Find **Column Filter**.
 - b. For Type, select how to derive the filters provided in the report column drop down list. Options include:
 - **None** - No column filter is provided for this report.
 - **Default Based on Column Type** - `STRING` and `NUMBER` columns derive the list of values from distinct column values. `DATE` columns derive the list of values from predefined date ranges.
 - **Use Defined List of Values to Filter Word Contains** - Derive the list of values from a defined list of values SQL query to create a contains filter.
 - **Use Named List of Values to Filter Exact Match** - Derive the list of values from the named list of values to create an equal filter.
 - **Use Named List of Values to Filter Word Contains** - Derive the list of values from the named list of values to create a contains filter.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Formatting Columns in an Interactive Report

You can use HTML expressions in interactive reports to further the customize column display.

To define column formatting:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.

The columns appear in the Rendering tab.

3. Select a column to edit.

The Property Editor displays the attributes for that column. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Configure the column format:
 - a. Find **Column Formatting**.
 - b. In HTML Expression, enter HTML expressions to be shown in this column. Use `#COLUMN#` syntax to show column values in HTML.

- **Example 1: Change the background and text colors**

```
<span style="background-color: yellow; color: red;">#EMPLOYEE_STATUS#</span>
```

- **Example 2: Use Heading 2**

```
<h2>#EMPLOYEE_NAME#</h2>
```

- **Example 3: Use a dynamic class from the SQL query**

```
<span class="#COLUMN_CLASS#">#EMPLOYEE_STATUS#</span>
```

The `#COLUMN_CLASS#` and `#EMPLOYEE_STATUS#` substitutions in this example are columns selected in the report SQL query.

- **Example 4: Show an image, and a tooltip on hover**

```

```

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

About Filtering on Conditional Links in Interactive Report Columns

If an interactive report has a column value that is a link and the link markup is constructed in the report query, attempting to define a filter on this value from the column header menu fails. Oracle Application Express is not able to render the link correctly in the column header menu because the column values are themselves rendered as links, such that they can be selected to apply the filter. If the developer wanted to conditionally render the link based on query logic, the link may have been defined in the report query as opposed to using the declarative column link support.

For filtering to work on conditional links in columns, developers have two options:

- **Option 1: Use an HTML Expression column:**
 1. Add another column to your query that selects the equivalent of the link text from the link and define a column alias of `my_link_display`. This column alias will be the value used in the column header menu.
 2. Change the `my_link` column to be of type **Hidden Column**.
 3. In the `my_link_display` column, set the column heading to be the same as the `my_link` column and set **HTML Expression** to be `#MY_LINK#`. Keep the default **Escape Special Characters** as **Yes**.
 4. Run the page.

If needed, reorder the columns so that the new column is in the same position as the old one. Select the **Actions** menu and then **Select Columns**.
 5. Save a new default report. Select the **Actions** menu, **Save Report**, and then **Save as Default Report Settings**.

The link still renders fine in the report and the column header menu now renders just the `my_link_display` value which sets the filter correctly.
- **Option 2: Use a custom Column Filter for the link column, where you could query for the link text.**



See Also:

["Editing Interactive Report Column Attributes "](#)

Customizing Interactive Reports in a Running Application

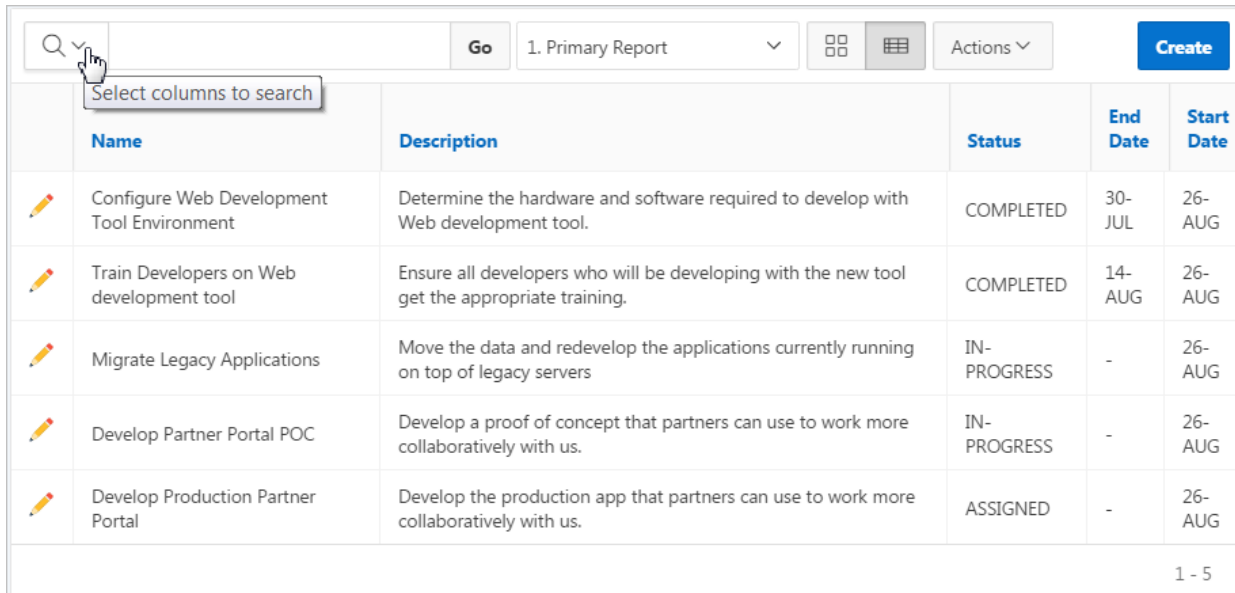
When running an application, users (that is, both end users and developers) can alter the layout of interactive reports data by choosing the columns they are interested in, applying filters, highlighting, and sorting. They can also define breaks, aggregations, charts, group bys, and add their own computations.






- [About the Search Bar](#)
- [About the Actions Menu](#)
- [Using Interactive Report Filters](#)
- [Saving Interactive Reports](#)

 **See Also:**

"Customizing an Interactive Report Using the Actions Menu" in *Oracle Application Express End User Guide*

About the Search Bar



| Name | Description | Status | End Date | Start Date |
|--|--|-------------|----------|------------|
|  Configure Web Development Tool Environment | Determine the hardware and software required to develop with Web development tool. | COMPLETED | 30-JUL | 26-AUG |
|  Train Developers on Web development tool | Ensure all developers who will be developing with the new tool get the appropriate training. | COMPLETED | 14-AUG | 26-AUG |
|  Migrate Legacy Applications | Move the data and redevelop the applications currently running on top of legacy servers | IN-PROGRESS | - | 26-AUG |
|  Develop Partner Portal POC | Develop a proof of concept that partners can use to work more collaboratively with us. | IN-PROGRESS | - | 26-AUG |
|  Develop Production Partner Portal | Develop the production app that partners can use to work more collaboratively with us. | ASSIGNED | - | 26-AUG |

1 - 5

A search bar displays at the top of every interactive report and may include the following features:

- **Select Columns to Search** - This icon resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
- **Text Area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go Button** - Executes a search.
- **Reports** - Displays alternate default and saved private, or public reports. See "[Saving Interactive Reports](#)."
- **View Icons** - Switches between an View Icon, View Report and View Detail of the default report (if enabled). May also include Chart and Group By View (if defined).
- **Actions menu** - Use the Actions menu to customize an interactive report. See "[About the Actions Menu](#)."

 **See Also:**

"Customizing the Interactive Report Search Bar"

About the Actions Menu

Note:

The structure of the interactive report Actions menu has changed to be consistent with interactive grids. To alter interactive report Actions menu to use the previous (or legacy) structure, go to **Shared Components, Component Settings**, and configure the **Interactive Report** component. See "[Configuring Interactive Report Action Menu Structure](#)".

The Actions menu appears to the right of the Go button on the Search bar. End users use the Actions menu to customize an interactive report.

| Name | Description | End Date | Start Date |
|--|--|----------|------------|
| Configure Web Development Tool Environment | Determine the hardware and software required to develop with Web development tool. | 0- JUL | 26- AUG |
| Train Developers on Web development tool | Ensure all developers who will be developing with the new tool get the appropriate training. | 4- AUG | 26- AUG |
| Migrate Legacy Applications | Move the data and redevelop the applications currently running on top of legacy servers | | 26- AUG |
| Develop Partner Portal POC | Develop a proof of concept that partners can use to work more collaboratively with us. | | 26- AUG |
| Develop Production Partner Portal | Develop the production app that partners can use to work more collaboratively with us. | | 26- AUG |

See Also:

"Customizing an Interactive Report Using the Actions Menu" in *Oracle Application Express End User Guide*

Using Interactive Report Filters

When you customize an interactive report, a filter displays between the search bar and the report. The following illustration shows an interactive report with one filter, `Status = 'COMPLETED'`.

| Name | Description | Status | End Date | Start Date |
|--|--|-----------|----------|------------|
| Configure Web Development Tool Environment | Determine the hardware and software required to develop with Web development tool. | COMPLETED | 30-JUL | 26-AUG |
| Train Developers on Web development tool | Ensure all developers who will be developing with the new tool get the appropriate training. | COMPLETED | 14-AUG | 26-AUG |

Other keys features of interactive report filters include:

- **Enable/Disable** - To enable and disable a filter, select and deselect the check box to the left of the filter name.
- **Filter Edit** - To edit a filter, click the filter name.
- **Remove Filter** - To remove a filter, select the **Remove Filter** icon to the right of the filter name.

By default, multiple filters display horizontally stacked on top of one another. To minimize the amount of information that displays and hide filter details, click the arrow to the left of the filter name. The following illustration shows the same report with the filter details hidden.

| Name | Description | Status | End Date | Start Date |
|--|--|-----------|----------|------------|
| Configure Web Development Tool Environment | Determine the hardware and software required to develop with Web development tool. | COMPLETED | 30-JUL | 26-AUG |
| Train Developers on Web development tool | Ensure all developers who will be developing with the new tool get the appropriate training. | COMPLETED | 14-AUG | 26-AUG |

Saving Interactive Reports

All users can save a private or public interactive report. However, only the user who creates a private report can view, save, rename, or delete it. Developers have additional save capabilities in that they can save the report that initially displays, called the Primary Default, or create an Alternative Default report.

- [About the User Type and Available Save Options](#)

- [About Configuration Dependencies](#)
- [Saving a Public or Private Interactive Report](#)
- [Saving a Default Interactive Report](#)
- [Renaming a Public or Private Interactive Report](#)
- [Deleting a Public or Private Interactive Report](#)
- [About Exporting Interactive Reports](#)



See Also:

"Managing Saved Interactive Reports" in *Oracle Application Express Administration Guide*

About the User Type and Available Save Options

An end user can save an interactive report in two ways:

- **Public Report** (End user and developer). This report is viewable by all users. However, only the user who creates a public report can save, rename, or delete it. Although all users can view a public report, they can only save it under a new report name.
- **Private Report** (End user and developer). Only the user who creates the private report can view, save, rename, or delete it.

A developer can additionally save an interactive report in the following ways:

- **Primary Default** (Developer only). The Primary Default is the report that initially displays. Primary Default reports cannot be renamed or deleted.
- **Alternative Report** (Developer only). Enables developers to create multiple report layouts. Only developers can save, rename, or delete an Alternative Report.

The following illustration shows the Reports list on the Search bar of an interactive report.

| | Name | Description | Status | End Date | Start Date |
|--|--|--|-------------|----------|------------|
| | Configure Web Development Tool Environment | Determine the hardware and software required to develop with Web development tool. | COMPLETED | 30-JUL | 26-AUG |
| | Train Developers on Web development tool | Ensure all developers who will be developing with the new tool get the appropriate training. | COMPLETED | 14-AUG | 26-AUG |
| | Migrate Legacy Applications | Move the data and redevelop the applications currently running on top of legacy servers | IN-PROGRESS | - | 26-AUG |
| | Develop Partner Portal POC | Develop a proof of concept that partners can use to work more collaboratively with us. | IN-PROGRESS | - | 26-AUG |
| | Develop Production Partner Portal | Develop the production app that partners can use to work more collaboratively with us. | ASSIGNED | - | 26-AUG |

This example shows two reports:

- **Default - Primary Report.** This is the initial report created by the application developer. Default, Primary reports cannot be renamed or deleted.
- **Private - 1. Projects with Tasks.** This is a Private report. Only the user who creates a private report can view, save, rename, or delete it.

 **Tip:**

You can view saved report activity from the Workspace Administration page. On the Workspace home page, click the **Administration menu** and then select **Administration, Monitory Activity** and **By Interactive Report**.

About Configuration Dependencies

Developers control how users can save an interactive report by editing report Attributes.

To fully enable users to save private reports, a developer must:

- Define an authentication scheme for the current application.
- Edit the following report Attributes:
 - **Actions Menu, Save Report** — Set this attribute to **Yes** to enable end users to save their current report definition as a *Public* report.
 - **Actions Menu, Save Public Report** — Set this attribute to **Yes** to enable end users to save their current report definition as a *Private* report.
 - **Search Bar, Reports Select List** — Set this attribute to **Yes** to include a Reports select list in the Search bar.
 - **Actions Menu, Save Public Report Authorization** — Select an authorization scheme to restrict who can save public report definitions. To enable an end user to save a public report, this authorization scheme must evaluate to TRUE. If no authorization scheme is provided then reports may be saved by any user.

 **See Also:**

- ["Establishing User Identity Through Authentication"](#)
- ["Customizing the Actions Menu"](#)
- ["Customizing the Interactive Report Search Bar"](#)

Saving a Public or Private Interactive Report

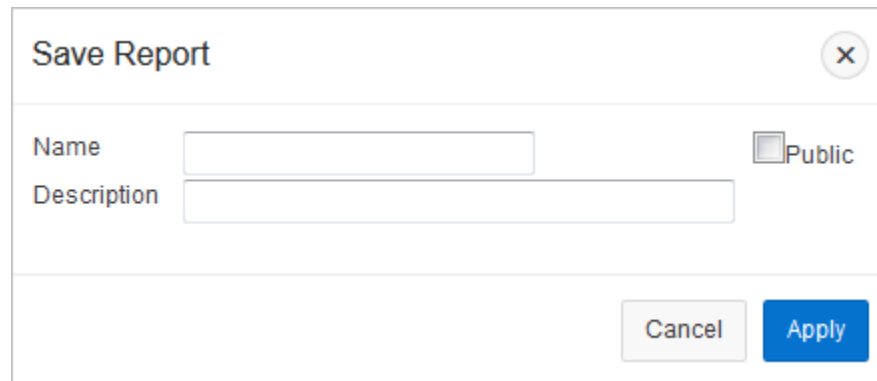
End users can save an interactive report and classify it as being either public or private. Public reports can be viewed by all users. Private reports can only be viewed by the user who saves the report.

 **Tip:**

To save a public report, the developer must edit the report Attributes. In the Property Editor, locate **Actions Menu** and set **Save Public Report** to **Yes**. Developers can also apply an authorization scheme to enable or disable Save Public Report. See "[About Configuration Dependencies](#)."

To save a public or private interactive report:

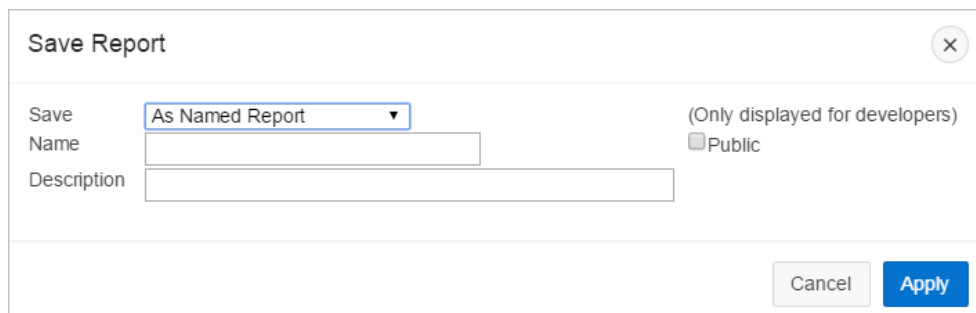
1. Go to the page containing the interactive report you want to save.
2. Customize the report (for example, hide columns, add filters, and so on).
3. Click the **Actions** menu and select **Report** and then **Save Report**.
The Save Report dialog appears.
4. For end users, the following Save Report dialog appears.



The dialog box is titled "Save Report" and has a close button (X) in the top right corner. It contains two text input fields: "Name" and "Description". To the right of the "Name" field is a checkbox labeled "Public". At the bottom right of the dialog are two buttons: "Cancel" and "Apply".

In Save Report:

- a. **Name** - Enter a name for the report.
 - b. **Public**- Select this check box to make the report viewable to all users. Deselect this check box to make the report private.
 - c. **Description** - Enter an optional description.
 - d. Click **Apply**.
5. For developers, the following Save Report dialog appears.



The dialog box is titled "Save Report" and has a close button (X) in the top right corner. It contains a dropdown menu labeled "Save" with "As Named Report" selected. Below the dropdown is a text input field for "Name". To the right of the "Name" field is a checkbox labeled "Public" and the text "(Only displayed for developers)". Below the "Name" field is a text input field for "Description". At the bottom right of the dialog are two buttons: "Cancel" and "Apply".

In Save Report:

- a. **Save** - Select **As Named Report**.
- b. **Name** - Enter a name for the report.
- c. **Public**- Select this check box to make the report viewable to all users. Deselect this check box to make the report private.

 **Tip:**

The Public check box only displays if you have enabled the **Save Public Report** report attribute.

- d. **Description** - Enter an optional description.
- e. Click **Apply**.

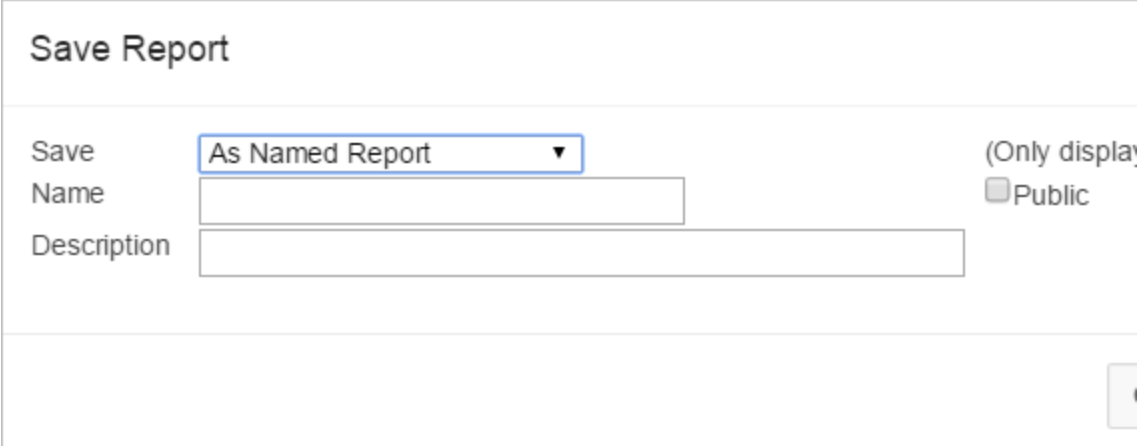
Saving a Default Interactive Report

Developers can save two types of default interactive report: **primary** and **alternative**. Both reports display on the Report list on the search bar. However, primary default reports cannot be renamed or deleted.

To save a default interactive report:

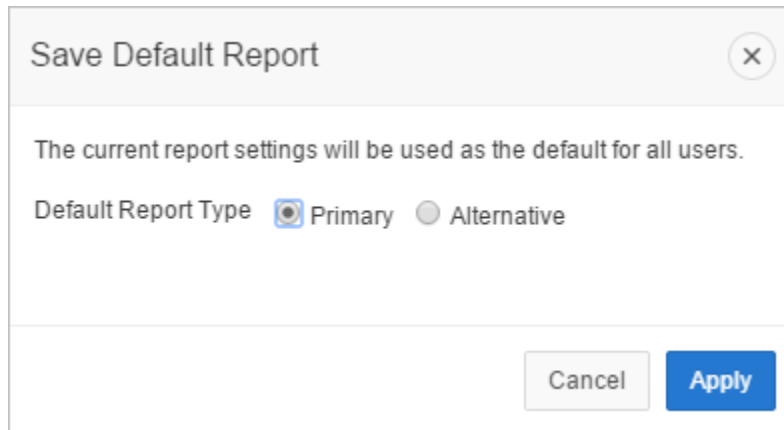
1. Run the report as a developer.
2. Customize the report (for example, hide columns, add filters, and so on).
3. Click the **Actions** menu and select **Report** and then **Save Report**.

The Save Report dialog appears.



4. From Save, select **As Default Report Settings**.

The Save Default Report dialog appears.

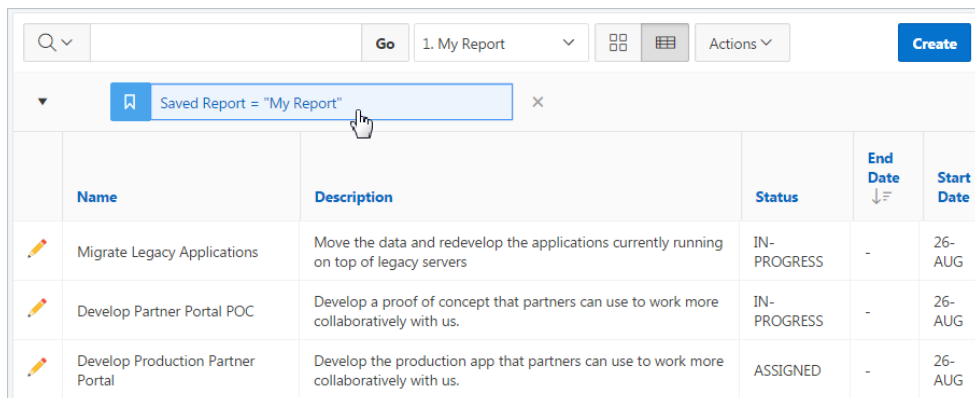


5. For Default Report Type, select either:
 - **Primary** - This report displays as the primary report when the page runs.
 - **Alternative** - This report displays as an alternative option on the Reports list on the Search bar. If you select **Alternative**, enter a name.
6. Click **Apply**.

Renaming a Public or Private Interactive Report

To rename a public or private interactive report:

1. Run the report as a developer.
2. Select the public or private interactive report to rename.
3. Click the saved report link.

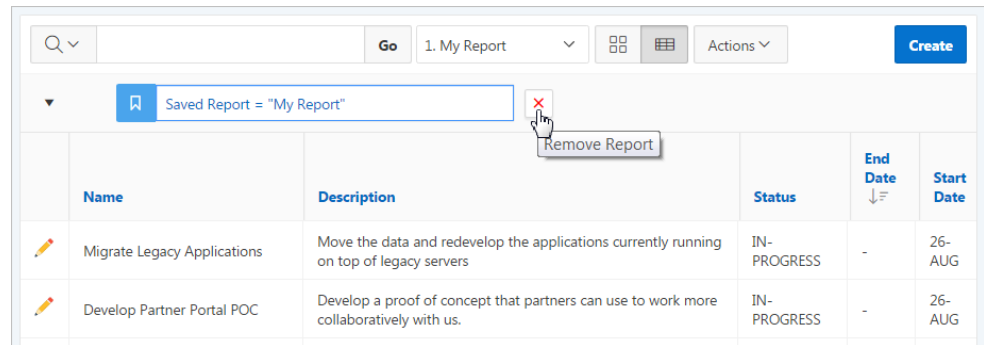


4. In the Rename Report dialog, edit the attributes (for example, enter a new name) and click **Apply**.

Deleting a Public or Private Interactive Report

To delete a public or private interactive report:

1. Run the report as a developer.
2. Click the **Remove Report** icon next to the report name link.



| Name | Description | Status | End Date | Start Date |
|-----------------------------|---|-------------|----------|------------|
| Migrate Legacy Applications | Move the data and redevelop the applications currently running on top of legacy servers | IN-PROGRESS | - | 26-AUG |
| Develop Partner Portal POC | Develop a proof of concept that partners can use to work more collaboratively with us. | IN-PROGRESS | - | 26-AUG |

3. Click **Apply**.

About Exporting Interactive Reports


When you export an application, the Primary Default, Alternative Default, public reports are included by default. You can also choose to include personal reports during the export process by selecting **Yes** for Private Interactive Reports.

 **See Also:**
["Exporting an Application"](#)

Linking to Interactive Reports

Developers can use the syntax in the `Request` value and `ItemNames` section of the URL to link to interactive reports.

Data in an interactive report is driven by declarative filters. Developer can set report parameters by modifying interactive report settings for an end user. However, these settings only affect the end user's working report and do not affect any saved report settings. Developers can use the syntax described in this section to link to interactive reports.

 **Note:**
 When you add additional interactive report regions, be sure to change the old single interactive report link syntax. Otherwise, the link generates an error. A link to a page with more than one interactive report regions, the request or name syntax needs to fully qualify with region static ID.

- [Request Syntax](#)
- [Name and Value Syntax](#)
- [ClearCache Syntax](#)
- [Restrictions When Linking to Interactive Reports](#)
- [Valid Linking Examples](#)

- [Invalid Linking Examples](#)
- [Linking to Shared Interactive Reports](#)

**See Also:**

["Understanding URL Syntax"](#) and ["About Using f?p Syntax to Link Pages"](#)

Request Syntax

To link to a specific saved report, use the following `Request` syntax:

```
IR[region static ID]<report_alias>
```

Consider the following example:

```
IR[EMP]_dept20
```

**Note:**

The use of `Request` in the URL is optional. To link to multiple saved reports, separate each `Request` with comma.

Name and Value Syntax

To create a filter, use the following `itemNames` and `itemValues` syntax:

```
IR[region static ID]<operator>_<target column alias>
```

Consider the following example:

```
IR[EMP]C_ENAME:KING
```

**Note:**

If there is only one interactive report region on the page, the `itemName` does not need to be fully qualified with the region static ID. If an operator is not defined, a filter with equal operator is added.

To create a filter on an interactive report in a link, use the `itemNames` syntax described previously and add the filter value in the corresponding location in the item names section of the URL.

To create row text which contains a filter on an interactive report, use `ROWFILTER` as the target column alias in the item name syntax and add a filter value in the `itemValues` of the URL. The link will add a Row Text Contains filter which searches all columns displayed in the report with type `STRING` or `NUMBER`.

Consider the following example:

```
IR[EMP]_ROWFILTER:SALES
```

Valid operators include:

- C = Contains
- EQ = Equals (this is the default)
- GTE = Greater than or equal to
- GT = Greater Than
- LIKE = SQL Like operator
- LT = Less than
- LTE = Less than or equal to
- N = Null
- NC = Not Contains
- NEQ = Not Equals
- NLIKE = Not Like
- NN = Not Null
- NIN = Not In (escape the comma separated values with backslash, \)
- IN = In (escape the comma separated values with backslash, \)
- ROWFILTER = Row Text Contains (this searches all columns displayed in the report with type STRING or NUMBER)

ClearCache Syntax

Tip:

Using ClearCache used without specifying a saved report alias in the `Request`, clears the primary default report. To clear a specific interactive report region on a page with more than one interactive reports, specify the region static ID along with saved report alias in the `Request`.

Use the following syntax in the `ClearCache` section of the URL to reset, clear, or reset pagination the interactive report:

- RIR - Resets an interactive report. This is equivalent to the end user choosing the Reset option from the interactive report Actions menu on the target page. The report is returned to the default report settings specified by the developer or saved by the user.
- CIR - Clears interactive report. This clears all of the report settings such as control break, aggregate, flashback, chart, number of rows to display, filter, highlight, computation, group by, and pivot.
- RP - Resets interactive report pagination.

Restrictions When Linking to Interactive Reports

When linking to interactive reports, remember the following restrictions:

1. If a `Request` value is not defined in the URL, all of the primary default report settings will change.
2. If a `Request` value is not defined and `itemNames` section does not include a fully qualified filter column name with region static ID, an error occurs if there are more than one interactive report region in the target page.
3. If a `Request` value is not defined and `itemNames` section do not include a fully qualified filter column name with region static ID, it will create a filter on the primary default report if there is only one interactive report region in the target page.

Valid Linking Examples

This section contains valid linking examples.

Valid Linking Example 1

This example includes a link with no `Request` value, but includes a qualified filter column with region static ID in the `itemNames` section of the URL. This example creates a filter on the primary default report in the referenced region item name.

```
f?p=&APP_ID.:2:&SESSION.:::IR[EMP]_ENAME:KING
```

Valid Linking Example 2

To support backwards compatibility with previous releases, this example uses older syntax and works on a page with one interactive report region.

```
?p=&APP_ID.:1:&SESSION.:IR_REPORT_12345::RIR,CIR::RIR,CIR:IR_ENAME:KING
```

Valid Linking Example 3

In this example the link has no `Request` value. The clear cache section has `RIR` or `CIR` or `RP` and will reset, clear, or reset the pagination of the primary default reports of all interactive report regions.

```
f?p=&APP_ID.:1:&SESSION.:::RIR:
```

Invalid Linking Examples

This section contains invalid linking examples.

Invalid Linking Example 1

This link has no `Request` value and does not include a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error if there is more than one interactive report region in the target page.

```
f?p=&APP_ID.:2:&SESSION.:::ENAME:KING
```

Invalid Linking Example 2

This example has a link with two request values separated by commas and does not contain a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error since it is not clear where the filter needs to be created.

```
f?p=&APP_ID.:2:&SESSION.:IR[test]_employee,IR[test2]_dept10:::ENAME:KING
```

Invalid Linking Example 3

This example uses old request value syntax used on a page with multiple interactive report regions. This example will generate an error. The old syntax only works if there is one interactive report region.

```
f?p=&APP_ID.:2:&SESSION.:IR_REPORT_myrrpt:::ENAME:KING
```

Linking to Shared Interactive Reports

You can link to saved primary default, alternative default, and public reports using `IR[region static ID]_<report_alias>` in the Request value of the URL.

- [Editing the Alias of Default Report in Page Designer](#)
- [Link Examples to a Primary Report](#)
- [Link Examples to Saved Reports](#)



See Also:

and

- ["About Using f?p Syntax to Link Pages"](#)
- ["Saving a Default Interactive Report "](#)
- ["Deleting Saved Interactive Reports"](#)
- ["Managing Saved Interactive Reports" in *Oracle Application Express Administration Guide*](#)

Editing the Alias of Default Report in Page Designer

You can view attributes for default interactive reports in the Property Editor.

To view attributes for a default interactive report:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Select the Saved Report attributes:

- a. In the Rendering tab, locate the region containing the report.
- b. Under Region, expand the **Attributes** node.
- c. Expand **Saved Reports**.
- d. Select a saved report.

The Saved Report attributes display in the Property Editor.

3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. **FindIdentification.**
5. In Alias, enter a new alias for this report. Use this alias to define a more meaningful URL link to this saved report. This link can be used to link directly from elsewhere in the application.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Link Examples to a Primary Report

The following examples demonstrate how to link to a primary report (report_alias=PRIMARY):

This example links to a primary report:

```
f?p=100:1:&APP_SESSION.:IR_PRIMARY
```

This example links, resets, and clears primary report settings:

```
f?p=100:1:&APP_SESSION.:IR_PRIMARY::RIR,CIR:
```

This example links, resets, and clears primary report settings. It additionally creates a ENAME = 'KING' filter on the primary report:

```
f?p=100:1:&APP_SESSION.:IR_PRIMARY::RIR,CIR:IR_ENAME:KING
```

Link Examples to Saved Reports

The following examples demonstrate how to link to a saved report (report_alias=12345):

This example links to a saved report:

```
f?p=100:1:&SESSION.:IR_REPORT1
```

This example links, resets, and clears settings for saved report REPORT1:

```
f?p=100:1:&SESSION.:IR_REPORT1::RIR,CIR:
```

This example links, resets, and clears saved report REPORT1 settings. It additionally creates a ENAME = 'KING' filter on saved report REPORT1.

```
f?p=100:1:&SESSION.:IR_REPORT1::RIR,CIR:IR_ENAME:KING
```

Managing Classic Reports

A classic report is the formatted result of a SQL query. Developers choose a table on which to build a report, or provide a custom SQL SELECT statement or a PL/SQL function returning a SQL SELECT statement. Developers control how a classic report works by editing the following attributes in Page Designer: region attributes, report Attributes, and Column attributes.

Tip:

With the exception of sorting and simple filtering, end users cannot customize a classic report.

- [Managing Classic Report Region Attributes](#)
Each application page contains one or more regions. A region is an area on a page that serves as a container for content. Developers edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.
- [Managing Classic Report Attributes](#)
Report Attributes control how a classic report works. Developers edit report Attributes to alter the report appearance, configure report pagination, create error messages, break formatting, and configure download options.
- [Managing Classic Report Column Attributes](#)
Column attributes enable developers to control the display, features, and column behavior. Developers edit Column attributes to alter nearly all aspects of column behavior, including altering the layout and appearance, defining column links, creating column filters, and adding support for export and printing.

Managing Classic Report Region Attributes

Each application page contains one or more regions. A region is an area on a page that serves as a container for content. Developers edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.

- [Editing Classic Report Region Attributes](#)

See Also:

["Editing Pages in Page Designer"](#)

Editing Classic Report Region Attributes

To edit region attributes in Page Designer:

1. View the page in Page Designer:

- a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
- Page Designer appears.
2. In the Rendering tab, select the region.
- The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

3. Edit the appropriate attributes.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

["Right Pane of Page Designer \(Property Editor\)"](#)

Managing Classic Report Attributes

Report Attributes control how a classic report works. Developers edit report Attributes to alter the report appearance, configure report pagination, create error messages, break formatting, and configure download options.

- [Editing Classic Report Attributes](#)
- [Editing Classic Report Pagination](#)
- [Adding Support for XML File or a CSV File Export](#)
- [Enabling Classic Report CSV Output Option](#)
- [Controlling Classic Report Column Breaks](#)

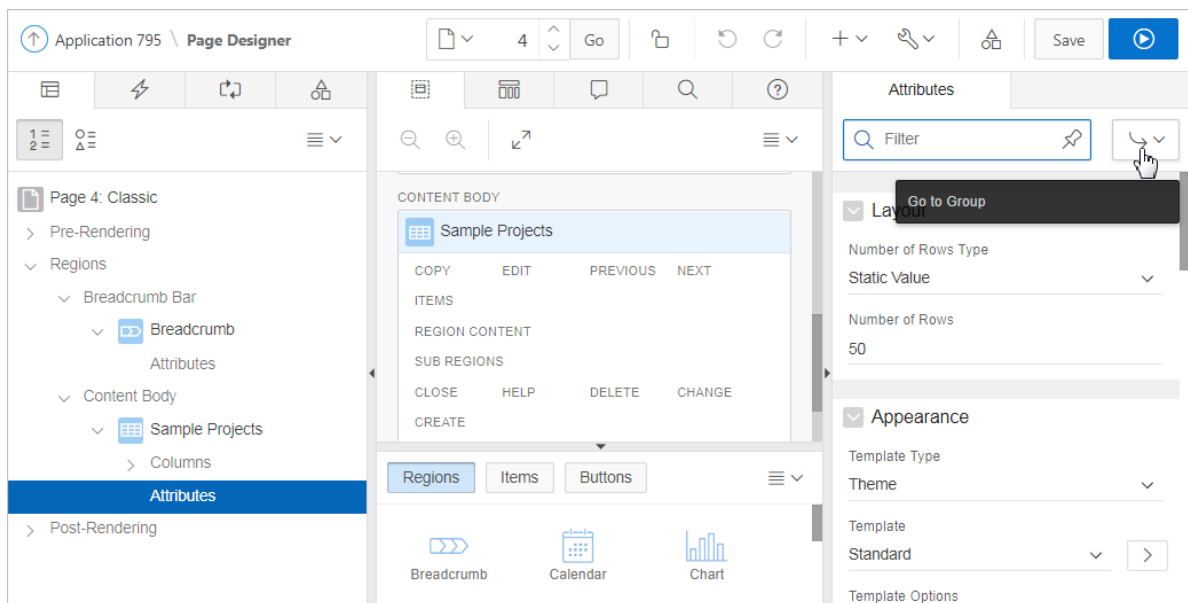
 **See Also:**["Editing Pages in Page Designer"](#)

Editing Classic Report Attributes

You can customize a classic report by editing report attributes in Page Designer.

To edit report attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
 Page Designer appears.
2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.



The Property Editor displays the report attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Edit the appropriate attributes.
5. To view Help, select the attribute and click the Help tab. Once you activate the Help pane, the content that displays changes every time you select another attribute.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Editing Classic Report Pagination

To control classic report pagination:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under Region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Find **Pagination**.
5. Under Pagination, edit the following:

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- a. **Type** - Select the pagination type for this report. Generally only a subset of the results are currently displayed in the report. Pagination provides the user with information about the number of rows and the current position within the result set. Pagination also defines the style of links or buttons that are used to navigate to the next or previous page.
- b. **Display Position** - Select where the report pagination displays.
- c. **Partial Page Refresh** - Specify whether to use Partial Page Refresh (PPR) to update the report display.

If the end user performs an action, such as a pagination request or selection of a new sort column, the current report rows must be refreshed. Setting this attribute to **Yes**, refreshes just the report region. Setting this attribute to **No**, refreshes the entire page.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Adding Support for XML File or a CSV File Export

You can add support for exporting a report as an XML or CSV by selecting a report template.

To export a report as a file:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Find **Appearance**.
5. Under Appearance, edit the following:
 - a. Template Type - Select **Predefined**.
 - b. Template - Select **XML** or **CSV**.

Selecting **XML** prevents the Application Express engine from rendering the page and dumps the content to an XML file.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Enabling Classic Report CSV Output Option

To use the enable the CSV output option:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.

- b. Select an application.
- c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the region, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Find **Download**.
5. Under Download, edit the following:
 - a. CSV Export Enabled - Specify whether end users can download the report contents to a CSV file. To enable CSV Downloads, you must use a report template with a #CSV_LINK# substitution string and set this option to **Yes**.
 - b. CSV Separator - Enter a column separator. If no value is entered, a comma or semicolon is used depending on your current NLS settings.
 - c. CSV Enclosed By - Enter a delimiter character to delineate the starting and ending boundary of a data value. The default delimiter is double quotation marks.
 - d. Link Text - Specify the text for the link to invoke the CSV download.
 - e. Filename - Specify a name for the CSV download file. If no name is specified, the region name is used followed by the extension *.csv.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Controlling Classic Report Column Breaks

You can control if a specific column repeats and how column breaks appear when printed using Break Formatting attributes. For example, suppose your report displays employee information by department number. If multiple employees are members of the same department, you can increase the readability by specifying that the department number only appears once.

To create this type of column break:

1. View the page in Page Designer:

- a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
Page Designer appears.
2. In the Rendering tab, locate the region containing the report.
 3. Under the region, select the **Attributes** node.
The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Specify Break Formatting:
 - a. Find **Break Formatting**.
 - b. Break Columns - Select how many report columns are incorporated into the break formatting. The columns must be consecutive columns, starting from the first column displayed in the report.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Managing Classic Report Column Attributes

Column attributes enable developers to control the display, features, and column behavior. Developers edit Column attributes to alter nearly all aspects of column behavior, including altering the layout and appearance, defining column links, creating column filters, and adding support for export and printing.

- [Altering Classic Report Column Layout](#)
- [Enabling Column Sorting in a Classic Report](#)
- [Creating a Column Link in a Classic Report](#)
- [Defining an Updatable Column in a Classic Report](#)
- [Defining a Column as a List of Values in a Classic Report](#)
- [Controlling When Classic Report Columns Display](#)
- [Adding a Download Link to a Classic Report](#)

 **See Also:**

["Editing Pages in Page Designer"](#)

Altering Classic Report Column Layout

You can use the Column Attributes section of the Report Attributes page to precisely control the report layout. For example, you can use these attributes to alter column heading text, change column positioning, hide a column, create a sum of a column, or select a sort sequence.

To edit the Column Attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.
3. Select a column to edit.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Edit the appropriate attributes in the Property Editor. Attributes are organized in groups.

The following table describes common ways to edit report columns.

Table 9-4 Common Report Column Edits for Classic Reports

| Description | Developer Action |
|---|--|
| Hide a column. | Find Identification . For Type, select Hidden Column . |
| Edit the column heading text. | Find Heading . Edit the Heading and Alignment attributes. |
| Change the column heading text alignment. | Find Heading . Under Heading, make a new selection from Alignment . |
| Alter the column display sequence. | Find Layout . Edit the Sequence attribute. |
| Enable a unique sort sequence. | Find Sorting . For Sortable select Yes and select a Default Sequence . Any number of columns can be sort enabled. However, at least one column must have a Sort Sequence defined. |
| Enable the sum of a column. | Find Advanced . For Compute Sum , select Yes . |

**Tip:**

To view Help, select the attribute and click the Help tab. Once you activate the Help pane, the content that displays changes every time you select another attribute.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Enabling Column Sorting in a Classic Report

To enable column sorting:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.
3. Select a column to edit.
The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Find **Sorting**.
6. Under Sorting, edit the following:
 - a. Default Sequence - Select the sort sequence for this column. Each column should have a unique sequence to correctly order the report output.
 - b. Sortable -To enable sorting, select **Yes**. Selecting Yes enables the end user to sort this column in the report.

**Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**Tip:**

Column sorting must be enabled if you want columns with null values to display at the top or end of the report. To learn more about the Sort Nulls attribute, see "[Editing Classic Report Pagination](#) ."

Creating a Column Link in a Classic Report

Use the Column Link attributes to create a link from a report to another page in your application or to a URL.

To create a column link to another page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.
3. Select a column to edit.
The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Find **Identification**. From Type, select **Link**.
6. Find **Link** and click **No Link Defined**.
The Link Builder - Target dialog appears.
7. To create a link to another page, in the Link Builder - Target dialog:
 - a. Type - Select **Page in this Application**.
 - b. Page - Specify the target page number.
 - c. Set Items - Select a Name and Value to specify session state for an item.
 - d. Clear Session State, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple page, enter a comma-delimited list of page numbers.
 - e. Reset Pagination - Select **Yes** to reset pagination for this page.
 - f. Advanced, Request - Specify the request to be used.
 - g. Click **OK**.

8. To create a link to a URL, in the Link Builder - Target dialog:
 - a. Type - Select **URL**.
 - b. URL - Enter the URL address.
 - c. Click **OK**.
9. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Defining an Updatable Column in a Classic Report

Developers can make a column updatable by editing column attributes. Note that the Application Express engine can only perform updates if:

- A multirow update is defined.
- A PL/SQL process is implemented to process updated data.
- When using the built-in tabular form elements and display types, then the report has to be defined using the type **SQL Query (updatable report)**.

To define updatable column attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.
3. Select a column to edit.
The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. In the Property Editor, find **Identification** and edit the Type attribute.

Tip:

To view Help, select the attribute and click the Help tab. Once you activate the Help pane, the content that displays changes every time you select another attribute.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Defining a Column as a List of Values in a Classic Report

A column can be rendered as Radio Group or a static, named, or query-based LOV. For example, a column can be rendered using a select list or a popup list of values. Or, a column can be rendered as read-only text based on a list of values.

This last approach is an effective strategy when creating display lookup values and is particularly useful in regular, nonupdatable reports. This approach enables you to display the value of a column without having to write a SQL JOIN statement.

To render a report column as a list of values:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.
The columns appear in the Rendering tab.
3. Select a column to edit.
The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Find **Identification**. From **Type**, select **Plain Text (based on List of Values)**.
6. Find **List of Values**. Select a **Type**:
 - **Shared Component** - Creates a list of values based on a predefined list of values defined in Shared Components.
 - **SQL Query** - Creates a dynamic list of values based on the SQL Query you provide.
 - **Static Values** - Creates a static list of values is based on the text value you provide.
 - **PL/SQL Function Body returning SQL Query** - Creates a dynamic list of values is based on the SQL Query returned by the entered PL/SQL Function Body you provide.
7. The UI changes based on the **List of Values, Type** you select. Fill in all required attributes (identified in red).
8. To save your changes click **Save**.
9. To save and run the page, click **Save and Run Page**.

Controlling When Classic Report Columns Display

You can use the Authorization Scheme and Server-side Condition attributes to control when a column displays.

Use Authorization Scheme to control access to resources (such as a report column) based on predefined user privileges. For example, you could create an authorization scheme in which only managers can view a specific report column. Before you can select an authorization scheme, you must first create it. Use Server-side Condition

A condition is a small unit of logic that enables you to control the display of a column based on a predefined condition type. The condition evaluates to true or false based on the values you enter in the Expressions fields.

To specify Authorization Scheme and Server-side Condition attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.
3. Select a column to edit.

The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. If controlling a column with an authorization scheme, select an authorization scheme:
 - a. Find **Security**.
 - b. Authorization Scheme - Make a selection.
6. If controlling the column with a condition define a condition, define a condition:
 - a. Find **Server-side Condition**.
 - b. Type - Make a selection.
 - c. Depending upon your selection, enter an expression or value in the fields provided.

If the authorization is successful and the condition type display evaluates to true, the column displays.
7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

- ["Providing Security Through Authorization"](#)
- ["Understanding Conditional Rendering and Processing"](#)
- ["Available Conditions"](#)

Adding a Download Link to a Classic Report

Developers can create a link within a report that enables users to export the report as a comma-delimited file (.csv) file. To add a CSV link to a report, you must enable **Automatic CSV Encoding**. When using Automatic CSV Encoding, the report template is not important. You can include a CSV link with any report template that has the CSV export substitution string defined.

 **See Also:**

["Accessing the Globalization Attributes Page"](#) and ["Automatic CSV Encoding"](#)

Printing Report Regions

Configure a report region to print by exporting it to several different formats.

 **Tip:**

If you run Oracle Application Express with Oracle Database 11g Release 1 (11.1) or later, you must enable network services to use report printing. See ["Enabling Network Services in Oracle Database 11g or Later."](#)

By taking advantage of region report printing, your application users can view and print reports that have a predefined orientation, page size, column headings, and page header and footer. Interactive reports also have the ability to export to PDF, RTF, Microsoft Excel and Comma Separated Values (CSV).

 **Note:**

Interactive reports do not support a custom report layout.

Oracle BI Publisher supports Adobe Portable Document Format (PDF), Microsoft Word Rich Text Format (RTF), or Microsoft Excel format (XLS), or Extensible Markup Language (XML). Oracle Rest Data Services also supports PDF and XML. If you

choose to use other third-party rendering engines, other output formats can also be configured.

- [About Printing Reports to PDF](#)
- [About Report Printing Configuration Options](#)
- [About Classic Report Printing Methods](#)
- [Configuring Classic Report Region Print Attributes](#)
- [Configuring Interactive Report Region Print Attributes](#)
- [About Report Queries](#)
- [About Report Layouts](#)

About Printing Reports to PDF

To print a report to PDF, the data must be transformed using a report server defined at the instance-level. From an end user's perspective, you simply clicks a print link. However, from a developer's perspective, you must declaratively create regions to support PDF printing.

About Report Printing Configuration Options

Oracle Application Express provides three report printing configuration options:

- **Oracle REST Data Services** - Select this option if you are using the Oracle REST Data Services (formerly called Application Express Listener) release 2.0 or later. This option enables you to use the basic printing functionality, which includes creating report queries and printing report regions using the default templates provided in Application Express and using your own customized XSL-FO templates.

Note:

The Oracle REST Data Services option does not require an external print server, instead the report data and style sheet are downloaded to the listener, rendered into PDF format by the listener and then sent to the client. The PDF documents in this setup are not returned back into the database, thus the print APIs are not supported when using the Oracle REST Data Services-based configuration.

- **External (Apache FOP)** - Select this option if you are using Apache FOP on an external J2EE server. This option enables you to use the basic printing functionality, which includes creating report queries and printing report regions using the default templates provided in Application Express and using your own customized XSL-FO templates.
- **Oracle BI Publisher** - This option requires a valid license of Oracle BI Publisher (also known as Oracle XML Publisher). This option enables you to take report query results and convert them from XML to RTF format using Oracle BI Publisher. Select this option to upload your own customized RTF or XSL-FO templates for printing reports within Application Express.

 **Tip:**

To learn more about installing and configuring Oracle BI Publisher, see "[About Report Printing Configuration Options](#)".

 **Note:**

To use the full functionality of report printing, your Oracle Application Express service administrator must enable it for your instance. See "Configuring Report Printing" in *Oracle Application Express Administration Guide*.

About Classic Report Printing Methods

There are two ways to print classic report regions:

- **Configure Printing Attributes for a Report Region.** You can print a report region by configuring the Printing Attributes for the region. See "[Configuring Classic Report Region Print Attributes](#)."
- **Create a Report Query.** You can print a report by defining a report query as a Shared Component. See "[Printing a Report Region by Defining a Report Query](#)."

Both report regions and report queries can be downloaded in the following formats:

- PDF - Adobe Portable Document Format.
- RTF - Microsoft Word Rich Text Format.
- XLS - Microsoft Excel format. Note that this is not a true `.xls` file because the content is HTML-based.
- HTML
- XML - Extensible Markup Language.

To format either a report region or report query, you associate it with a report layout.

 **Tip:**

You can also have the output format specified by an item that determines the output format at runtime. Select the item that holds the format value. Valid values are PDF, RTF (to open the document in Microsoft Word), XLS (to open the document in Microsoft Excel) and HTML (to download the document as an HTML file).

 **See Also:**

"[Formatting a Report Region or Report Query Using Report Layouts](#)"

Configuring Classic Report Region Print Attributes

One approach to printing a report region is to configure Print Attributes. Once configured, these attributes apply only to the current region and cannot be used outside the context of the region. If the printing feature is set up for your instance, you can configure a report region to print in various formats.

- [Configuring a Classic Report Print Attributes](#)

See Also:

"[About Classic Report Printing Methods](#)" and "Configuring Report Printing" in *Oracle Application Express Administration Guide*

Configuring a Classic Report Print Attributes

To configure a classic report region for printing:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under the report, select the **Attributes** node.

The Property Editor displays the report attributes for the page. Attributes are organized in groups.

Tip:

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Enable printing:
 - a. In the Property Editor, find the **Printing** group.
 - b. Set **Enabled** to **Yes**.
5. In the Rendering tab, expand the **Attributes** node and select the **Printing** node. Printing attributes appear in the Property Editor.
6. Edit the Printing attributes.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Configuring Interactive Report Region Print Attributes

Unlike classic reports, the interactive report Print Attributes can only use the default XSL-FO layout and is initiated from the Report Attributes, Download section rather than directly from this screen. Once configured, these attributes only apply only to the current region and cannot be used outside the context of the region.

If the printing feature is set up for your instance, you can configure a report region to print in various formats.

- [Configuring Interactive Report Regions for Printing](#)

Configuring Interactive Report Regions for Printing

To configure an interactive report region for printing:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In the Rendering tab, locate the region containing the report.
3. Under Regions, select the **Attributes** node.

The Property Editor displays the attributes. Attributes are organized in groups.

 **Tip:**

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Under the **Attributes**, select the **Printing** node.
The Printing Attributes display in the Property Editor.
5. Edit the appropriate Printing attributes.

**Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

About Report Queries

You can print a report region by defining a report query as a Shared Component. A report query identifies the data to be extracted. Unlike SQL statements contained in regions, report queries contain SQL statements that are validated when you save the query. Note that report queries must be SQL statements, not functions returning SQL statements.

**See Also:**

["Printing a Report Region by Defining a Report Query"](#)

About Report Layouts

To format either a classic report region or report query, you associate it with a report layout. Using report layouts renders the data in a printer-friendly format. If you do not select a report layout, a default XSL-FO layout is used. The default XSL-FO layout is always used for rendering Interactive Report regions.

**See Also:**

["Formatting a Report Region or Report Query Using Report Layouts"](#)

Understanding BLOB Support in Forms and Reports

Oracle Application Express includes declarative BLOB support to enable developers to declaratively upload files in forms, and download or display files in reports.

BLOB display and download can also be authored procedurally using PL/SQL. This section describes how to upload, download and display files, and how to manage additional file attributes such as MIME type and file name that are important for proper management of files stored in BLOB columns. Using this functionality you can easily extend your Oracle Application Express applications to manage files including images, documents, videos, and so on.

- [About BLOB Support in Reports](#)
- [About Providing a Download Link](#)

- [About Displaying the BLOB](#)
- [About Working With BLOBs Procedurally](#)

About BLOB Support in Reports

Oracle Application Express includes BLOB support for both classic and interactive reports. If you use a wizard to create a report and include a column of type BLOB, basic support is included. Additional information should be added after generation to make the download capability more user friendly.

About Providing a Download Link

To facilitate the inclusion of a download link in a report, the report includes the selection of the length of the BLOB (for example, `dbms_lob.getlength(RESUME)`). If the length is 0, the BLOB is NULL and no download link is displayed. In the same way you specify a format mask for a date or number you can format a download link. The DOWNLOAD format is more complex however than other format masks in that you are required to specify at least three parameters, for example:

```
DOWNLOAD:EMP:RESUME:EMPNO
```

The following table describes the parameters of the DOWNLOAD format:

| Position | Attribute | Required | Description |
|----------|------------------------|----------|---|
| 1 | DOWNLOAD | Yes | Identifies the DOWNLOAD report format mask. |
| 2 | Table Name | Yes | Case sensitive name of table containing target column of type BLOB. |
| 3 | Column containing BLOB | Yes | Case sensitive name of column of type BLOB. |
| 4 | Primary Key Column 1 | Yes | Case sensitive name of primary key column 1. |
| 5 | Primary Key Column 2 | No | Case sensitive name of primary key column 2. |
| 6 | MIME type Column | No | Case sensitive column name used to store the MIME type. |
| 7 | Filename Column | No | Case sensitive column name used to store the filename of the BLOB. If NULL, the column name is used as the default when a user downloads the file. |
| 8 | Last Update Column | No | Case sensitive column name used to store the last update date of the BLOB. If used, the HTTP header of the file download indicates the date of last modification and web browsers will be able to cache the BLOB. If not specified, the browser may not be able to cache files. |
| 9 | Character Set Column | No | Case sensitive column name used to store the character set of the BLOB. Most relevant for Asian languages which may need to maintain the character set encoding. |

| Position | Attribute | Required | Description |
|----------|---------------------|----------|--|
| 10 | Content Disposition | No | Specify <code>inline</code> or <code>attachment</code> . All other values are ignored. If a <code>MIME</code> type is provided and the file is a type that can be displayed, the file is displayed. If <code>MIME</code> type is not provided, or the file cannot be displayed inline, the user is prompted to download. |
| 11 | Download Text | No | String used for the download link. If nothing is provided, <code>Download</code> is used. Note that this supports substitutions (useful for translated applications). |
| 12 | Table Owner | Yes | Identifies name of table owner containing target column of type BLOB. |

Consider the following example:

```
DOWNLOAD:EMP:RESUME:EMPNO::RESUME_MIMETYPE:RESUME_FILENAME:RESUME_LAST_UPDATE::attachment:Resume
```

To be able to enter these parameters and create the format as described above. You have to select **Blob Format** for Number/Date format item. Once selected, the Blob Column attributes region displays and you can fill in all parameters as described above.

About Displaying the BLOB

If the BLOB you are working with is an image, you can display it in the report using the new report format mask of 'IMAGE'. Regardless of the `MIME` type, the report always attempts to display the BLOB. If the BLOB cannot be rendered, a broken image is displayed.

The following table describes the parameters of the IMAGE format mask:

| Position | Attribute | Required | Description |
|----------|------------------------|----------|--|
| 1 | IMAGE | Yes | Identifies the IMAGE report format mask. |
| 2 | Table Name | Yes | Case sensitive name of table containing target column of type BLOB. |
| 3 | Column containing BLOB | Yes | Case sensitive name of column of type BLOB. |
| 4 | Primary Key Column 1 | Yes | Case sensitive name of primary key column 1. |
| 5 | Primary Key Column 2 | No | Case sensitive name of primary key column 2. |
| 6 | MIME type Column | No | Case sensitive column name used to store the <code>MIME</code> type. |
| 7 | Filename Column | No | Not used for IMAGE format but left in so that the format can easily be changed between IMAGE and DOWNLOAD. |

| Position | Attribute | Required | Description |
|----------|----------------------|----------|---|
| 8 | Last Update Column | No | Case sensitive column name used to store the last update date of the BLOB. If used, the HTTP header of the file download indicates the date of last modification and web browsers will be able to cache the BLOB. If not specified, the browser may not be able to cache files. |
| 9 | Character Set Column | No | Not used for IMAGE format but left in so that the format can easily be changed between IMAGE and DOWNLOAD. |
| 10 | Content Disposition | No | Not used for IMAGE format but left in so that the format can easily be changed between IMAGE and DOWNLOAD. |
| 11 | Alt Text | No | String used for the alt tag associated with the image. |
| 12 | Table Owner | Yes | Identifies name of table owner containing target column of type BLOB. |

Consider the following example:

```
IMAGE:EMP:RESUME:EMPNO::RESUME_MIMETYPE:RESUME_FILENAME:RESUME_LAST_UPDATE::attachment:Resume
```

To be able to enter these parameters and create the format as described above. You have to select Blob Format for Number/Date format item. Once selected, a new region, Blob Column attributes, displays and you can fill in all parameters as described above.

About Working With BLOBs Procedurally

As an alternative to using the built-in methods of providing a download link, you can use the `APEX_UTIL.GET_BLOB_FILE_SRC` function. One advantage of this approach, is the ability to more specifically format the display of the image (with height and width tags). Note that this approach is only valid if called from a valid Oracle Application Express session. Also, this method requires that the parameters that describe the BLOB to be listed as the format of a valid item within the application. That item is then referenced by the function.



See Also:

"GET_BLOB_FILE_SRC Function" in the *Oracle Application Express API Reference*

10

Developing Forms

Developers can create a variety of different application forms either manually or by using wizards. Some forms enable users to update a single row in a table, while other forms enable users to update multiple rows in a table.

- [Understanding Form Types](#)
When creating a database application, developers can create three basic form types: editable interactive grid, form on a table, or a master detail.
- [Creating a Form](#)
Developers can create forms using a number of wizards including the Create Application Wizard, Create Page Wizard, and Create Region Wizard.
- [About Making an Interactive Grid Editable](#)
Developers determine whether the underlying data is read-only or editable by users. An interactive grid presents users a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.
- [Managing Master Detail Forms](#)
Developers can create a master detail form with either the Create Application Wizard or the Create Page Wizard.
- [Validating User Input in Forms](#)
Developers can create validation and an associated error message to check the data a user enters before processing.
- [Understanding BLOB Support in Forms](#)
Oracle Application Express includes BLOB support to enable developers to declaratively upload and download files in forms.

See Also:

- ["Creating a Legacy Master Detail Form"](#)
- ["Managing Legacy Tabular Forms"](#)
- ["Developing Reports"](#)

Understanding Form Types

When creating a database application, developers can create three basic form types: editable interactive grid, form on a table, or a master detail.

- [Editable Interactive Grid](#)
An interactive grid presents users with a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.

- [Form on a Table](#)
Form on a Table creates a form that enables users to update a single row in a database table.
- [Master Detail Forms](#)
A master detail form reflects a one-to-many relationship between two tables in a database. Master detail forms enable users to insert, update, and delete values from two tables or views.



See Also:

["Creating a Legacy Master Detail Form"](#)

Editable Interactive Grid

An interactive grid presents users with a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.

Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse. The following is an example of an editable interactive grid.

| <input type="checkbox"/> | | Project Lead | Name | Description | Created | Completed ↑= |
|-------------------------------------|--|-----------------|----------------------------------|-----------------------------------|-------------|--------------|
| <input checked="" type="checkbox"/> | | Lucille Beatie | Configure Web Development ... | Determine the hardware and s... | 04-APR-2018 | 06-JAN-2018 |
| <input type="checkbox"/> | | Lucille Beatie | Train Developers on Web deve... | Ensure all developers who will... | 04-APR-2018 | 17-JAN-2018 |
| <input type="checkbox"/> | | Lucille Beatie | Develop New Reporting Apps | Develop apps to meet C Level... | 04-APR-2018 | 31-JAN-2018 |
| <input type="checkbox"/> | | Bernard Jackman | Develop IT Management Apps | Develop apps to allow IT to m... | 04-APR-2018 | 11-FEB-2018 |
| <input type="checkbox"/> | | Lucille Beatie | Develop Customer Tracker Ap... | Develop an application to trac... | 04-APR-2018 | 03-MAR-2018 |
| <input type="checkbox"/> | | Bernard Jackman | Implement Customer Satisfacti... | Implement an application to t... | 04-APR-2018 | 03-MAR-2018 |



See Also:

["Using an Editable Interactive Grid"](#) in *Oracle Application Express End User's Guide*

Form on a Table

Form on a Table creates a form that enables users to update a single row in a database table.

The following is an example of a form on table.

The screenshot shows a form titled "Form on a Table" with the following fields and controls:

- * Name**: A text input field with a required field asterisk.
- Description**: A text input field.
- Project Lead**: A text input field.
- * Completed**: A date picker field with a required field asterisk.
- * Created**: A date picker field with a required field asterisk.
- * Updated**: A date picker field with a required field asterisk.
- Cancel**: A button located at the bottom left.
- Create**: A blue button located at the bottom right.



See Also:

[Creating a Form Using the Create Page Wizard](#)

Master Detail Forms

A master detail form reflects a one-to-many relationship between two tables in a database. Master detail forms enable users to insert, update, and delete values from two tables or views.

Typically, a master detail form displays a master row and multiple detail rows within a single HTML form. Developers can create a single page or two page master detail. You choose the tables on which to build the master and detail regions. Master Detail form options include:

- **Stacked** - Creates a single page master detail with editable interactive grids.
- **Side by Side** - Creates a single page (or Side by Side) master detail with a master table and detail table. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report.
- **Drill Down** - Creates a two page (or Drill Down) master detail. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grid for the detail.

 **See Also:**

- ["Managing Master Detail Forms"](#)
- ["About Master Detail Forms"](#)

Creating a Form

Developers can create forms using a number of wizards including the Create Application Wizard, Create Page Wizard, and Create Region Wizard.

- [Creating a Form Using the Create Page Wizard](#)
Use the Create Page Wizard to wide variety of forms.
- [Form Options When Running the Create Page Wizard](#)
Learn about form options when running the Create Page Wizard.
- [Creating a Form Using the Create Application Wizard](#)
Use the Create Application Wizard to create a form.
- [Creating a Form Region](#)
Use the Create Form Region Wizard to create a complex form region which includes multiple form items and processes.

 **See Also:**

- ["Developing Reports"](#)
- ["Managing Legacy Tabular Forms"](#)
- ["Creating a Legacy Master Detail Form"](#)

Creating a Form Using the Create Page Wizard

Use the Create Page Wizard to wide variety of forms.

To create a form using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page.
This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then select either **Form** or **Master Detail**.

**Tip:**

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

- c. If you select **Form**, select a form type:
 - **Report with Form on Table**
 - **Report with List View on Table** (Optimized for mobile apps)
 - **Editable Interactive Grid**
 - **Form on a Table**
 - **Form on a Procedure**
 - **Form on a SQL Query**
 - **Form on Web Service**
 - **Form on a SQL Query**
 - **Form on a Web Service**
 - **Report and Form on Web Service**
 - d. If you select **Master Detail**, select a type:
 - **Stacked** - Creates single page master detail with editable interactive grids. With this page, users can query, insert, update, and delete values from two tables or views which are related. You choose the tables on which to build the master and detail regions.
 - **Side by Side** - Create a single page master detail with a master table and detail table. In addition, the page includes a searchable region on the left that displays a primary and secondary column from the master table. Use this region to select a record to display.
 - **Drill Down** - Creates a two page master detail creates one interactive grid on the first page for the master table and on the second page it creates a standard form (for the master) and interactive grid for the detail.
 - e. Click **Next**.
5. What appears next depends upon the form you select. Follow the on-screen instructions.

To learn more about an attribute, see field-level Help.

**See Also:**

- ["Form Options When Running the Create Page Wizard"](#)
- ["Managing Master Detail Forms"](#)

Form Options When Running the Create Page Wizard

Learn about form options when running the Create Page Wizard.

Tip:

Universal Theme - 42 is optimized to work equally well in either a mobile or desktop environment. In previous releases and in earlier themes, the Mobile user interface is based on jQuery Mobile. Oracle recommends migrating existing mobile applications to the Universal Theme as soon as possible.

Available Forms in the Create Page Wizard

The Create Page Wizard supports the following forms.

Table 10-1 Create Page Wizard — Available Forms

| Form Type | Description | To Learn More |
|----------------------------------|---|---|
| Report with Form on Table | Creates report and form on a table you select. You select the report type (Interactive Report , Interactive Grid , or Class Report). One page displays as an interactive grid. Each row provides a link to the form to enable users to update each record. Note: This wizard does not support tables having more than 127 columns. Selecting more than 127 columns generates an error. | Not applicable. |
| Report with a List View on Table | Optimized for mobile applications. Creates a form and list view that enables users to update a single row in a database table. You choose the table on which to build the form and select the column to be used for displaying text in the List View. | Not applicable |
| Editable Interactive Grid | Creates an editable interactive grid based on a table you select or a SQL query. An interactive grid presents users a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse. | See " Managing Interactive Grids " and " Making an Existing Interactive Grid Editable " |
| Form on a Table | Creates a form that enables users to update a single row in a database table. You can choose a table on which to build a form. | Not applicable. |

Table 10-1 (Cont.) Create Page Wizard — Available Forms

| Form Type | Description | To Learn More |
|--------------------------------|--|--|
| Form on a Procedure | Builds a form based on stored procedure arguments. Use this approach when you have implemented logic or Data Manipulation Language (DML) in a stored procedure or package. | Not applicable. |
| Form on a SQL Query | Creates a form based on the columns returned by a SQL query such as an <code>EQUIJOIN</code> . | Not applicable. |
| Form on Web Service | Creates a page with items based on a Web service definition. This wizard creates a user input form, a process to call the Web service, and a submit button. | See "Creating a Form on a Web Service" |
| Report and Form on Web Service | Creates a page with items based on a Web service definition. This wizard creates a user input form, a process to call the Web service, a submit button, and displays the results returned in a report. | See "Creating an Input Form and Report on a Web Service" |

Available Master Detail Options in the Create Page Wizard

The Create Page Wizard supports the following master detail options.

Table 10-2 Create Page Wizard — Available Master Detail Options

| Form Type | Description | To Learn More |
|--------------|---|---|
| Stacked | Creates a single page (or Stacked) master detail with editable interactive grids. Users can query, insert, update, and delete values from two tables or views which are related. You choose the tables on which to build the master and detail regions. | See "Creating a Stacked Master Detail Using the Create Page Wizard" |
| Side by Side | Creates a single page (or Side by Side) master detail with a master table and detail table. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report | See "Creating a Side by Side Master Detail Using the Create Page Wizard" |
| Drill Down | Creates a two page (or Drill Down) master detail form. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grid for the detail. | See "Creating a Two Page Drill Down Master Detail Using the Create Page Wizard" |

Available Forms for Mobile Applications

In previous releases, developers selected the **Mobile** User Interface to optimize applications for mobile environments. The Mobile user interface is based on jQuery Mobile. If your application uses an older theme and the Mobile User Interface, the Create Page Wizard supports the following forms.

Table 10-3 Create Page Wizard — Forms for Mobile Applications

| Form Type | Description | To Learn More |
|----------------------------------|--|--|
| Report with a List View on Table | Creates a form and list view that enables users to update a single row in a database table. You choose the table on which to build the form and select the column to be used for displaying text in the List View. | Not applicable |
| Form on a Table | Creates a form that enables users to update a single row in a database table. You can choose a table on which to build a form. | Not applicable |
| Form on a Procedure | Builds a form based on stored procedure arguments. Use this approach when you have implemented logic or Data Manipulation Language (DML) in a stored procedure or package. | Not applicable |
| Form on a SQL Query | Creates a form based on the columns returned by a SQL query such as an <code>EQUIJOIN</code> . | Not applicable |
| Form on Web Service | Creates a page with items based on a Web service definition. This wizard creates a user input form, a process to call the Web service, and a submit button. | See "Creating a Form on a Web Service" |
| Report and Form on Web Service | Creates a page with items based on a Web service definition. This wizard creates a user input form, a process to call the Web service, a submit button, and displays the results returned in a report. | See "Creating an Input Form and Report on a Web Service" |

**See Also:**

["Creating a Form Using the Create Page Wizard"](#)

Creating a Form Using the Create Application Wizard

Use the Create Application Wizard to create a form.

To create a form using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Click **New Application**.
4. For Name, enter the name used to identify the application to developers.
5. For Appearance, accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.

6. To add a form, click **Add Page** and select then select **Form**.
7. On Create Form Page:
 - a. Page Name - Enter a name for this page.
 - b. Set Icon - Select an icon to display in the navigation menu for this page.
 - c. Table - Select the table for the form.
 - d. Include Report - Select **Include Report** to create a report and form. The first page is an interactive report. Each row provides a link to a form where users can update a record.
 - e. Advanced - Expand **Advanced**:
 - Page Help - Enter text to be displayed when the user selects Page Help.

 **Tip:**

This setting requires you select the **About Page** feature. If you select the **About Page** feature, a help icon is generated in the navigation bar with an entry for page help.

- f. Click **Add Page**.
8. Under Features, select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.

 **Tip:**

Click the **Check All** button to select all features.

9. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
10. Click **Create Application**.

 **See Also:**

"Creating Database Applications"

Creating a Form Region

Use the Create Form Region Wizard to create a complex form region which includes multiple form items and processes.

 **Tip:**

Using the Create Form Region Wizard creates a complex form region which includes multiple form items and processes. Although you can create a region directly in the Rendering tree or drag a region from the Gallery, these approaches will not create these additional components.

To create a form using the Create Form Region Wizard:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Locate the Page Designer Toolbar at the top of the page.
3. On Page Designer Toolbar, click the **Create** menu and select **Form Region**.
The Create Region Wizard appears.
4. On Create Form Region:
Options include:
 - a. Select the type of form you wish to create. Options include:
 - **Form on a Table**
 - **Form on a SQL Query**
 - **Form on a Procedure**
 - **Form on a Web Service**
 - b. Click **Next**.
5. What appears next depends upon the form region. Follow the on-screen instructions.

About Making an Interactive Grid Editable

Developers determine whether the underlying data is read-only or editable by users. An interactive grid presents users a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.

As described in "Using Interactive Grids" in *Oracle Application Express End User's Guide*, users can lock, hide, filter, freeze, highlight, and sort individual columns. Advanced users can also define breaks, aggregations, and computations against

columns. Users can also directly customize the appearance of an interactive grid. Users can use the mouse to resize the width of a column and drag and drop columns into different places in the grid.

In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page. Editable interactive grids expand the functionality of regular interactive grids to enable users to more directly update a grid's structure and contents. Users can add and delete rows, edit cell contents, and refresh the grid with the latest updates.

 **See Also:**

["Making an Existing Interactive Grid Editable"](#) and ["Managing Interactive Grids"](#)

Managing Master Detail Forms

Developers can create a master detail form with either the Create Application Wizard or the Create Page Wizard.

- [About Master Detail Forms](#)
- [Creating a Master Detail Form Using the Create Application Wizard](#)
- [Creating a Side by Side Master Detail Using the Create Page Wizard](#)
- [Creating a Stacked Master Detail Using the Create Page Wizard](#)
- [Creating a Two Page Drill Down Master Detail Using the Create Page Wizard](#)

 **See Also:**

["Creating Master Detail from an Existing Interactive Grid"](#) and ["Creating a Legacy Master Detail Form"](#)

About Master Detail Forms

A master detail form reflects a one-to-many relationship between tables in a database. Typically, a master detail form displays a master row and multiple detail rows. When you create a master detail form using the Create Page or Create Application Wizards, you choose the tables on which to build the master form and the detail form. Master detail form options include: Stacked, Side by Side, and Drill Down.

 **Tip:**

To see master detail examples, install the sample application, *Sample Master Detail* and select a master detail type in the left navigation menu. See ["Installing a Productivity and Sample App."](#)

Stacked Master Detail

A **Stacked** master detail form contains a single page with editable interactive grids based on two related tables or views. Users select a row in the master grid to update the detail grids. Developers can create a stacked master detail with either the Create Application Wizard or the Create Page Wizard.

The following example displays master detail interactive grids on the same page. The user has option of interacting with either the master or the detail without leaving the page.

| | | Project Details | | Project Details | | Progress | Associated Records | | |
|-------------------------------------|---|------------------------------------|--|-----------------|-------------|------------|--------------------|----------|--|
| <input type="checkbox"/> | ≡ | Name | Description | Project Lead | Status | Milestones | Tasks | Comments | |
| <input type="checkbox"/> | ≡ | Configure Web Development Too... | Determine the hardware and softwa... | Tyson King | Completed | 0 | 6 | 3 | |
| <input type="checkbox"/> | ≡ | Train Developers on Web develop... | Ensure all developers who will be d... | Lucille Beatie | Completed | 2 | 6 | 2 | |
| <input type="checkbox"/> | ≡ | Migrate Legacy Applications | Move the data and redevelop the a... | Miyazaki Yo... | In-Progress | 3 | 9 | 6 | |
| <input checked="" type="checkbox"/> | ≡ | Develop Production Partner Portal | Develop the production app that pa... | Lucille Beatie | Assigned | 5 | 17 | 0 | |
| <input type="checkbox"/> | ≡ | Develop Partner Portal POC | Develop a proof of concept that par... | Tyson King | In-Progress | 2 | 7 | 0 | |

1 rows selected 1 - 5

The previous illustration shows the master interactive grid with the project, *Develop Production Partner Portal* selected. The next illustration shows detail interactive grids for *Milestones* and *Tasks*.

| | | Milestones | | | | | | | |
|-------------------------------------|---|-------------------------------|---|-------------|-------|--|--|--|--|
| <input type="checkbox"/> | ≡ | Name | Description | Due Date ↑↕ | Tasks | | | | |
| <input checked="" type="checkbox"/> | ≡ | Define Production App Sc... | Based on the results of the POC, define the requirements for the production ... | 27-SEP-18 | 3 | | | | |
| <input type="checkbox"/> | ≡ | Build Phase 1 of Productio... | Develop the modules defined in the first phase of the application. | 04-OCT-18 | 3 | | | | |
| <input type="checkbox"/> | ≡ | Perform Beta testing with ... | Work with a few key partners to trial Phase 1 of the Partner Portal app. | 12-OCT-18 | 3 | | | | |
| <input type="checkbox"/> | ≡ | Complete Phase 1 Develo... | Based on the results of the Beta program, enhance the application to make p... | 08-NOV-18 | 4 | | | | |
| <input type="checkbox"/> | ≡ | Roll out Phase 1 of Partne... | Go-Live for the Partner Portal application to all partners. | 12-NOV-18 | 4 | | | | |

1 rows selected Total 5

| | | Tasks | | | | | | | |
|-------------------------------------|---|--|-------|---------------|-----------|--------------|----------------------------|----------------------|--|
| <input type="checkbox"/> | ≡ | Name | A... | Start Date ↑↕ | End Date | Is Complete? | Milestone | Description | |
| <input checked="" type="checkbox"/> | ≡ | Define production scope of Partner ... | Ty... | 26-SEP-18 | 27-SEP-18 | No | Define Production App S... | Define the scope and | |
| <input type="checkbox"/> | ≡ | Finalize Partner App Data Model | D... | 26-SEP-18 | 27-SEP-18 | No | Define Production App S... | Refine the data mod | |
| <input type="checkbox"/> | ≡ | Finalize User Experience | M... | 26-SEP-18 | 27-SEP-18 | No | Define Production App S... | Write developer stan | |

Drill Down Master Detail

A **Drill Down** master detail contains two pages based on two related tables or views. The first page contains an interactive report for the master table. The second page

features a standard form for the master and interactive grids for the detail. Developers can create a Drill Down master detail with the Create Page Wizard.

| <input type="text" value="Q"/> <input type="button" value="Go"/> <input type="button" value="Actions"/> <input type="button" value="Create Project"/> | | | | | |
|---|--|-------------------|-------------|----------------|--|
| | Name | Project Lead | Status | Completed Date | Description |
| | Configure Web Development Tool Environment | Tyson King | Completed | 18-AUG-18 | Determine the hardware and software required to develop with Web development tool. |
| | Train Developers on Web development tool | Lucille Beatie | Completed | 02-SEP-18 | Ensure all developers who will be developing with the new tool get the appropriate training. |
| | Migrate Legacy Applications | Miyazaki Yokohama | In-Progress | - | Move the data and redevelop the applications currently running on top of legacy servers |
| | Develop Partner Portal POC | Tyson King | In-Progress | - | Develop a proof of concept that partners can use to work more collaboratively with us. |
| | Develop Production Partner Portal | Lucille Beatie | Assigned | - | Develop the production app that partners can use to work more collaboratively with us. |

1 - 5

In this example, users click the Edit icon to view project details. The second page includes both a standard form (for the master) and editable interactive grids.

Drill Down \ Develop Partner Portal POC

< > 4 of 5

[Show All](#) [Project Details](#) [Milestones](#) [Tasks](#) [Comments](#)

Project Details

Name *

Project Lead

Status *

Description

Audit Details

Milestones

Search: All Text Columns

| <input type="checkbox"/> | <input type="checkbox"/> | Name | Due Date ↑ | Description | Tasks |
|-------------------------------------|--------------------------|------------------------|------------|--|-------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | Define Requirements | 19-SEP-18 | Work with key stakeholders to define the scope of the project, and design ... | 3 |
| <input type="checkbox"/> | <input type="checkbox"/> | Build Proof-of-Concept | 26-SEP-18 | Create the initial screens and populate with data so key stakeholders can r... | 4 |

For numeric and date fields, you can also pre-define date and number format masks, or apply those format masks after generating the initial form. Wizard generated master detail forms also automatically create validations for some columns. Validations are created for columns that are set to NOT NULL in the underlying table and columns of type NUMBER, DATE or TIMESTAMP. Note that validations are not created for columns if the column is set to read-only, either based on the user's selection, or defined user interface defaults.



Tip:

You can control the appearance of a specific column by editing column attributes.

Side by Side Master Detail

A **Side by Side** master detail form contains a single page master detail utilizing side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report(s). Developers can create Side by Side master detail with either Create Application Wizard or Create Page Wizard.

Side by Side

↶ Reset
+ Add Project

Configure Web Development Tool Environment

COMPLETED

Develop Partner Portal POC

IN-PROGRESS

Develop Production Partner Portal

ASSIGNED

Migrate Legacy Applications

IN-PROGRESS

Train Developers on Web development tool

COMPLETED

▼ Overview: Side by Side

This example displays master-detail regions on the same page. The user selects a record from the master on the left to render the detail on the right without leaving the page.

Project Details

✎ Edit Project

| | |
|--------------|--|
| Project Lead | Lucille Beatie |
| Name | Develop Production Partner Portal |
| Description | Develop the production app that partners can use to work more collaboratively with us. |
| Status | Assigned |
| Created | 9/14/2018 |
| Created By | ADMIN |
| Updated | 9/14/2018 |
| Updated By | ADMIN |

The illustration that follows shows the Milestones detail interactive report. Users can Milestones by clicking an Edit icon.

ORACLE

10-14

| | | | | | | | | |
|--|--|--|--|-------------------------------|----------------|-------------------|----------------|-------------------|
| Side by Side | Created | 9/14/2018 | Reset | + Add Project | | | | |
| <input type="text" value="Search..."/> | Created By | ADMIN | | | | | | |
| Configure Web Development Tool Environment COMPLETED | Updated | 9/14/2018 | | | | | | |
| | Updated By | ADMIN | | | | | | |
| Develop Partner Portal POC IN-PROGRESS | Show All | Milestones | Tasks | Comments | | | | |
| Develop Production Partner Portal ASSIGNED | Milestones + | | | | | | | |
| Migrate Legacy Applications IN-PROGRESS | | Name | Description | Due Date | Created | Created By | Updated | Updated By |
| Train Developers on Web development tool COMPLETED | | Define Production App Scope | Based on the results of the POC, define the requirements for the production app. | 27-SEP-18 | 9/14/2018 | ADMIN | 9/14/2018 | ADMIN |
| | | Build Phase 1 of Production Partner Portal App | Develop the modules defined in the first phase of the application. | 04-OCT-18 | 9/14/2018 | ADMIN | 9/14/2018 | ADMIN |
| | | Perform Beta testing with select Partners | Work with a few key partners to trial Phase 1 of the Partner Portal app. | 12-OCT-18 | 9/14/2018 | ADMIN | 9/14/2018 | ADMIN |

Creating a Master Detail Form Using the Create Application Wizard

Create a master detail that enables users to query, insert, update, and delete values from two related tables or views. You choose the tables on which to build the master and detail regions.

Tip:

The Create Application Wizard uses general best practices to generate the application and does not include several advanced options included in the Create Page Wizard.

To create a master detail form using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Click **New Application**.
4. For Name, enter the name used to identify the application to developers.
5. For Appearance, accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
6. To add a report, click **Add Page** and select then select **Master Detail**.
7. On Add Master Detail Page:

- a. Page Name - Enter a name for this page.
- b. Set Icon - Select an icon to display in the navigation menu for this page.
- c. Page Type - Select one of the following:
 - **Side by Side** - Creates a single page (or Side by Side) master detail with a master table and detail table. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report.
 - **Stacked** - Creates single page master detail with editable interactive grids.

If Page Type is **Side by Side**:

- Table - Select the master table for the page.
- Primary Display Column - Select the column to display in the list of master records.
- Secondary Display Column - Select the descriptive column to display in the list of master records, under the Primary Display Column.
- Detail Table - Select the detail table.

If Page Type is **Stacked**:

- Table - Select the master table for the editable interactive grid.
- Detail Table - Select the detail table for the page.

- d. Advanced - Expand **Advanced**:
 - Page Help - Enter text to be displayed when the user selects Page Help.

 **Tip:**

This setting requires you select the **Help Pages** feature. If the **Help Pages** feature is not selected, the Help icon is not be generated and end users cannot access page help.

- e. Click **Add Page**.
8. Under Features, select features to include with the application. Features provide application-level functionality and are typically added once per application. To learn more, click the Help icon adjacent to **Features**.

 **Tip:**

Click the **Check All** button to select all features.

9. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
10. Click **Create Application**.
11. To view the new master detail page, click **Run Page**.

The rendered page appears.

**See Also:**

["Creating Database Applications"](#)

Creating a Side by Side Master Detail Using the Create Page Wizard

Use the Create Page Wizard to create a single page master detail with a master table and detail table. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report(s).

To create a Side by Side master detail using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Master Detail**.

**Tip:**

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

5. On Create Page, select **Side by Side**.
6. On Page Attributes:
 - a. Starting Page Number - Enter an unused starting page number. This wizard will create multiple pages starting with the number specified. The default is the first available page number within this application.
 - b. Master Detail Page Name - Enter the name of the master detail page which contains side by side master detail regions. The left side contains a master list to navigate to navigate to the master record. The right side contains the selected master record and the associated detail report.
 - c. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page. If you elect to include breadcrumb navigation, additional attributes appear.
 - d. Click **Next**.
7. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.

8. For Master Detail Source:
 - a. Table/View Owner - Select the owner of the table on which you are building master detail page.
 - b. Master Table - Select the master table or view.
 - c. Primary Display Column - Select the column to display as primary column in the report to navigate to a different master record.
 - d. Secondary Display Column - Select the column to display as secondary column in the report to navigate to a different master record.
 - e. Detail Table - Select the detail table or view.
 - f. Detail Table 2 - Select the detail table or view (optional).
 - g. Detail Table 3 - Select the detail table or view (optional).
 - h. Detail Table 4 - Select the detail table or view (optional).
 - i. Click **Create**.
9. To view the new master detail, click **Save and Run Page**.
The rendered pages appear.

Creating a Stacked Master Detail Using the Create Page Wizard

Use the Create Page Wizard to create a single page (or Stacked) master detail form. A Stacked master detail form features two editable interactive grids based on two related tables or views.

To create a single page master detail using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page.
This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Master Detail**.

Tip:


Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

5. On Create Page, select **Stacked**.
6. On Page Attributes:
 - a. Page Number - Enter the page number of the master page. If you identify a new page number, the wizard creates a new page. If you identify an existing page number, the wizard adds the component to that page.

Creating a Two Page Drill Down Master Detail Using the Create Page Wizard

Use the Create Page Wizard to create a two page (or Drill Down) master detail form. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grid for the detail.

To create a two page master detail using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
 2. Select an application.
 3. Click **Create Page**.
 4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Master Detail**.
-  **Tip:**
Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.
5. On Create Page, select **Drill Down**.
 6. On Page Attributes:
 - a. Master Page Number - Enter the page number of the master page. The master page will contain a read-only master interactive grid. Enter a new page number or an existing page number.
 - b. Master Page Name - Enter the name of the master page.
 - c. Detail Page Number - Enter the page number of the detail page. The detail page will contain a form and editable interactive grid..
 - d. Detail Page Name - Enter the name of the detail page.
 - e. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page. If you elect to include breadcrumb navigation, additional attributes appear.
 - f. Click **Next**.
 7. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
 8. For Master Source:

- a. Table/View Owner - Select the owner of the table on which you are building the interactive grid.
 - b. Table/View Name - Select the table or view on which the grid is based.
 - c. Primary Key Column - Select the name of the column that uniquely identifies a row in the table.
 - d. Form Navigation Order - Select the navigation order column used by the previous and next buttons on the Detail Page which navigates to a different master record.
 - e. Select Columns - Select one or more columns to be included in the region. Select all columns to be displayed.
 - f. Click **Next**.
9. For Detail Source:
- a. Table/View Owner - Select the owner of the table on which you are building the interactive grid.
 - b. Table/View Name - Select the table or view on which the grid is based.
 - c. Primary Key Column - Select the name of the column that uniquely identifies a row in the table.
 - d. Master Detail Foreign Key - Select the foreign key that relates between the master and detail tables.
 - e. Select Columns - Select one or more columns to be included in the region. Select all columns to be displayed.
 - f. Click **Create**.
- Page Designer appears.
10. To view the new master detail, click **Save and Run Page**.
- The rendered pages appear.

Validating User Input in Forms

Developers can create validation and an associated error message to check the data a user enters before processing.

- [About Creating Validations and Error Messages](#)
- [Associating a Validation with a Specific Item](#)
- [About the Error Message Display Location](#)

About Creating Validations and Error Messages

You can use validations to check data a user enters before processing. Once you create a validation and the associated error message, you can associate it with a specific item. You can choose to have validation error messages display inline (that is, on the page where the validation is performed) or on a separate error page.

Creating an inline error message involves these steps:

- Step 1 - Create a validation and specify error message text. To learn more, see "[Understanding Validations](#)."

- Step 2 - Associate the validation with a specific item.

Associating a Validation with a Specific Item

To associate an item with a validation and specify error message text:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In Page Designer, select the Processing tab in the left pane.
3. Under Validating, select the validation you want to associate.
The Property Editor displays Validation attributes.
4. Under Validation, edit the following attributes:
 - Error, Error Message - Enter the text to be displayed in the event that the validation does not pass.
 - Error, Display Location - Select where the error message displays for this validation. Validation error messages display on a separate error page, or inline with the existing page. Inline error messages display underneath the associated item label or in a notification area defined as part of the page template.
 - Error, Associated Item - Select the item where this validation error message displays. .

Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save**.

About the Error Message Display Location

The error message display location identifies where a validation error message displays. Validation error messages can display on an error page or inline within the existing page. Inline error messages can display in a notification area (defined as part of the page template) or within the field label.

To create a hard error that stops processes, including any remaining validations, you must display the error on an error page.

Understanding BLOB Support in Forms

Oracle Application Express includes BLOB support to enable developers to declaratively upload and download files in forms.

- [About BLOB Support in Forms](#)
- [About Uploading and Downloading Files into a BLOB](#)
- [About Displaying the BLOB](#)
- [About Removing the Image Reference](#)

About BLOB Support in Forms

If you create a form using the Create Application Wizard, create a page of type of **Form** or **Report and Form**, create a region of type **Form**, or add an item to an existing form, any item whose source is a database column of type BLOB results in an item of type **File Browse**. When the form is called for INSERT, the file selected by the user is loaded into the BLOB column. When the form is called for update, a download link is displayed to the right of the Browse button. Users can use this link to download the file.

About Uploading and Downloading Files into a BLOB

The defaulted BLOB support does not give you all the information a typical application needs to effectively manage a BLOB. In addition to knowing that the column is a BLOB, more information about the file provides a better experience for the end-user. The File Browse page item has additional settings to facilitate managing this additional information completely declaratively.

Tip:

For more information on File Browse settings, see **File Browse** in "[About Item Types](#)."

There are two different types of storage types available within the File Browse item type:

- BLOB column specific in Item Source Attribute - Completely declarative approach that supports configuration of the additional settings discussed here. This references a BLOB in your own database table.
- Table WWV_FLOW_FILES - Available for backwards compatibility. Oracle does not recommend using this in new applications.
- Table APEX_APPLICATION_TEMP_FILES - Store the uploaded file in a temporary location that you can access with the view APEX_APPLICATION_TEMP_FILES. Oracle Application Express automatically deletes the file at the end of the session or at the end of the upload request, depending on what you choose for Purge File At.

To provide this additional information, it is recommended that you add additional columns to your base table to store and track the MIME type, file name, last updated date and character set settings. You can accomplish this by extending your table. For example:

```
ALTER TABLE emp ADD
  (ATTACH_MIMETYPE    VARCHAR2(255),
   ATTACH_FILENAME    VARCHAR2(255),
   ATTACH_LAST_UPDATE DATE,
   ATTACH_CHARSET     VARCHAR2(128));
```

 **Note:**

The character set of the BLOB is not automatically set on upload. To store the character set value for your BLOB, you must provide an additional page item on your page which is bound to the column you use to store the character set, and where the user will be able to specify the character set for the document they are uploading.

If you manually create a form on a custom table, you can still take advantage of this feature. To do so, use the File Browse item type with a Storage Type setting of BLOB column specified in Item Source Attribute, on a page with a DML Process type of DML_PROCESS_ROW. This process determines the table name and primary key columns.

About Displaying the BLOB

If the BLOB you are working with is an image, you can display it in a form. To handle it declaratively, use the Display Image item type as described in ["About Item Types."](#) To handle it procedurally, see ["Understanding BLOB Support in Forms and Reports."](#)

About Removing the Image Reference

Because there is no set to NULL when using File Browse, if you need to provide a mechanism to remove an image reference, you must include a special Remove Image button to nullify the necessary columns. Consider the following example:

```
UPDATE demo_product_info
  SET product_image = NULL,
      MIMETYPE = NULL,
      FILENAME = NULL,
      IMAGE_LAST_UPDATE = NULL,
      CHARSET = NULL
 WHERE product_id = :P6_PRODUCT_ID;
```

11

Managing Database Application Components

Oracle Application Express supports the creation the following database application components: calendars, maps, charts, and application Help.

- [Understanding Oracle JET Integration with Oracle Application Express](#)
Oracle Application Express includes the Oracle JET (JavaScript Extension Toolkit) library. This section provides information about JET and how it is integrated with Oracle Application Express.
- [Switching from AnyChart to JET Chart](#)
Convert chart pages from the legacy AnyChart format (which may no longer render in apps) to Oracle JET Chart. Oracle recommends upgrading to JET Chart wherever possible.
- [Creating Calendars](#)
App Builder includes support for creating a calendar with monthly, weekly, daily, and list views. Once you specify the table on which the calendar is based, you can create drill-down links to information stored in specific columns and enable drag and drop capability.
- [Creating Maps](#)
App Builder includes built-in wizards for generating map charts on pages with a Desktop user interface. How you create a map chart depends upon whether you are adding the map to an existing page, or adding a map on a new page.
- [Creating Charts](#)
You can add a chart by running a built-in wizard or creating it manually in Page Designer.
- [Creating Help for Your Application](#)
Help created in App Builder displays on a dedicated Help page. To access Help, users click a link that takes them to a dedicated Help page. This Help page displays page and field-level Help topics specific to the page they are viewing.



See Also:

["Controlling Page Layout"](#) and ["Adding Navigation"](#)

Understanding Oracle JET Integration with Oracle Application Express

Oracle Application Express includes the Oracle JET (JavaScript Extension Toolkit) library. This section provides information about JET and how it is integrated with Oracle Application Express.

- [About Oracle JET](#)
- [How Oracle JET is integrated with Oracle Application Express](#)

About Oracle JET

Oracle JET (JavaScript Extension Toolkit) is a collection of open source JavaScript libraries with a set of Oracle contributed JavaScript libraries. Oracle JET is targeted at intermediate and advanced JavaScript developers and provides a rich set of UI components that can be easily integrated into Oracle Application Express. Oracle JET is a modular toolkit which enables developers to use as much or as little of the features that they desire.



See Also:

"[Oracle JET Homepage](#)" to learn more about Oracle JET.

How Oracle JET is integrated with Oracle Application Express

Currently Oracle Application Express integrates some parts of Oracle JET , primarily with `Chart` region types, and `Text field` with `Autocomplete` item types.



Note:

Although Oracle JET provides two-way data binding (using `knockout.js`), currently Oracle Application Express does not natively utilize this aspect of the toolkit.

Oracle JET uses a module loader (`RequireJS`) to only load modules that are required for specific functionality to work. This means that you do not have to load the entire Oracle JET library for certain functionality and also enables each module to define its own dependencies. At runtime, when a module is requested, the module loader looks at the module being requested for any dependencies to other modules. If there are any dependencies, these will also be loaded.

Module loader can however result in a large number of additional resources being individually loaded at runtime, which can have a negative performance impact. For this reason, Oracle Application Express makes use of the `RequireJS Optimization Tool`, which:

- provides a way to determine all the dependencies for modules either at design time or as part of a standard build process
- bundles them up into one combined, minified file

`RequireJS Optimization Tool` is loaded at runtime to avoid the need for all the additional requests.

Find the list of the bundles used for `Chart` and `Text field` with `Autocomplete` native types.

- `jetCommonBundle.min.js`
- `inputSearchBundle.min.js`
- `chartBundle.min.js`

 **Note:**

You can utilize the Common bundle `jetCommonBundle.min.js` in your plug-ins. This Common Bundle gets included on any Oracle Application Express page as soon as a file is included with your plug-in, which defines the `[require jet]` prefix. This means that any module your plug-in uses that is already included in the common bundle does not need to be refetched by `requireJS`.

`jetCommonBundle.min.js`

- **Directory:** `/images/libraries/apex/minified/`
- **Included when:** When a JavaScript file is requested that defines itself as requiring JET, by using the `[require jet]` prefix in Oracle Application Express, and the page is not in Debug mode

When debugging, the bundle is not loaded and `requireJS` loads all the individual resources separately.

`inputSearchBundle.min.js`

- **Directory:** `/images/libraries/apex/minified/`
- **Included when:** A Text Field with Autocomplete item is included on a page, and the page is not in debug mode.

When debugging, the bundle is not loaded and `requireJS` loads all the individual resources separately.

`chartBundle.min.js`

- **Directory:** `/images/libraries/apex/minified/`
- **Included when:** A JET chart is included on a page (for example, when a Chart region, or Interactive Grid region is added to a page), and the page is not in debug mode.

When debugging, the bundle is not loaded and `requireJS` loads all the individual resources separately.

 **See Also:**

- "[requireJS homepage](#)" to learn more about RequireJS
- "[RequireJS Optimization Tool](#)" to learn more about RequireJS Optimization Tool

Switching from AnyChart to JET Chart

Convert chart pages from the legacy AnyChart format (which may no longer render in apps) to Oracle JET Chart. Oracle recommends upgrading to JET Chart wherever possible.

In a previous release, the AnyChart Flash `.swf` files were removed from the `/images` folder. This results in your charts no longer rendering. No error message displays and the region is blank.

- [Upgrading to JET Chart Using Application Upgrade Utility](#)
Use the Application Upgrade utility to quickly upgrade AnyChart components in an application to JET Chart.
- [Searching a Workspace for AnyChart Using SQL Commands](#)
Use SQL Commands in SQL Workshop to search a workspace for AnyChart components. Upgrade them to JET Chart or implement a workaround by manually installing a `.zip` file.

Upgrading to JET Chart Using Application Upgrade Utility

Use the Application Upgrade utility to quickly upgrade AnyChart components in an application to JET Chart.

To use the Application Upgrade utility:

1. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
2. Select an application.
3. Click **Utilities**.
4. Click **Upgrade Application**.
5. Under Upgrade Type, review the list for one of the following messages:
 - "Upgrade AnyChart Charts to Oracle JET Charts"
 - "Upgrade Flash Charts to HTML5 Charts"
6. If either message appears, click the adjacent number in the Candidate Objects column to view pages containing these components.
7. Select objects to convert using the check box column and click the **Upgrade** button (top-right).

A success message appears, and the selected objects update and disappear from the list.

Searching a Workspace for AnyChart Using SQL Commands

Use SQL Commands in SQL Workshop to search a workspace for AnyChart components. Upgrade them to JET Chart or implement a workaround by manually installing a `.zip` file.

To locate legacy AnyChart components using SQL Commands:

1. On the Workspace home page, click the **SQL Workshop** icon.

The SQL Workshop home page appears.

2. Click **SQL Commands**.

Proceed through one or more of the following steps to scan your workspace for AnyChart content:

3. (Optional) To identify pages that have **AnyGantt charts**:

a. Input the following SQL statement:

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

b. Click **Run**.

4. (Optional) To identify pages that have **AnyChart maps**:

a. Input the following SQL statement:

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

b. Click **Run**.

5. (Optional) To identify pages that have **Flash-based AnyChart components**:

a. Input the following SQL statement:

```
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')
```

b. Click **Run**.

If your searches find no results, no further action is required.

6. If you locate pages with AnyChart content, do one of the following:

- (Recommended) Convert Flash-based AnyChart charts to Oracle JET (see ["Upgrading to JET Chart Using Application Upgrade Utility"](#)).
- As a workaround, manually install AnyChart Flash content to your instance by downloading the [AnyChart Flash .zip file](#) and following the contained `README.txt`. This populates the files necessary to render legacy AnyChart

content in your apps. Note that this workaround may not be an option in future releases. Upgrade to Oracle JET Chart to avoid further issues.

Creating Calendars

App Builder includes support for creating a calendar with monthly, weekly, daily, and list views. Once you specify the table on which the calendar is based, you can create drill-down links to information stored in specific columns and enable drag and drop capability.

- [About Supported Calendars](#)
App Builder supports two types of calendars: Calendar and Legacy Calendar.
- [Sample Calendar](#)
You can view a sample calendar by installing the sample application, *Sample Database Application* or the *Sample Calendar* application.
- [Adding a Calendar to a New Page Using a Local Database](#)
Run the Create Page Wizard to create a new page with a calendar using a local database.
- [Adding a Calendar on a New Page Using Remote Database References](#)
Run the Create Page Wizard to add a new page with a calendar that uses a remote database reference.
- [Creating a Calendar in Page Designer](#)
Manually create a calendar in Page Designer by providing the SQL query.
- [Managing Calendar Attributes](#)
Once you create a calendar, you can alter the display by editing attributes in the Property Editor. Each calendar has two sets of editable attributes: region attributes and calendar Attributes.
- [About Dynamic Action Support for Calendar](#)
Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. Using the Dynamic Action Create wizard, you specify an action that is performed when a defined set of conditions occur. You can also specify which elements are affected by the action, and when and how they are affected.



See Also:

["Managing Legacy Calendars"](#) and ["Integrating with Google Calendar"](#)

About Supported Calendars

App Builder supports two types of calendars: Calendar and Legacy Calendar.

Calendar is based on the FullCalendar jQuery library and can only be customized through CSS. **Legacy Calendar** is template-based and may be deprecated in the future release.

 **Tip:**

In previous releases **Legacy Calendar** was referred to as **Calendar (Template)**, **Classic calendar**, **Easy calendar**, and **SQL calendar**.

Both **Calendar** and **Legacy Calendar** enable you to create a calendar based on a table or SQL query you provide. During the creation process, you are prompted to select a date column and display column. Because it offers a better user interface and numerous additional features, Oracle recommends converting all legacy calendars to the latest calendar.

Calendar

The Calendar option supports:

- Rendering calendar events on multiple views (Month, Week, Day, or List).
- Rendering of duration and non-duration based events.
- Ability to modify the start and end dates by dragging and dropping events on different dates. Drag and drop is only supported for local data sources, that is, database objects in the referenced database schema and not on external data sources such as a Google calendar.
- Ability to change the duration by resizing the length of the events, by changing the end date.
- Ability to Edit or Adding new events on calendar using forms by clicking either on events or empty calendar cell.
- Rendering events from external sources using web service calls or Google Calendar feeds.
- Ability to render events titles on single or multiple lines.
- Using different CSS classes, developer can choose different styles for different type of events.
- Ability to download events in PDF printable format
- Support of multiple formats for sharing events (iCal, CSV, XML).
- Ability to add tooltip on Month/Week/Day views to make it easier for users to have a quick look at details of each event.

Legacy Calendar

Legacy Calendars option supports:

- Creation of one legacy calendar per page.
- Rendering of start date only.
- Look and feel defined by templates.
- Modification of start date by dragging and dropping events.
- Calendars that include daily, weekly, and monthly views.
- The date column determines the days on the calendar which contains entries.
- The display column defines a specific row which displays the calendar.

**See Also:**

["Managing Legacy Calendars"](#)

Sample Calendar

You can view a sample calendar by installing the sample application, *Sample Database Application* or the *Sample Calendar* application.

The following is a sample calendar from *Sample Database Application*.

| Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | |
|--------|--------|---------|-----------|----------|--------|----------|---|
| 25 | 26 | 27 | 28 | 29 | 30 | 1 | ▲ |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 | ▼ |

Features of this calendar include:

- Previous - Navigates to the previous month.
- Next - Navigates to the next month.
- Month - Displays a monthly view of all orders.
- List - Displays a list of all orders.

**Tip:**

To see more calendar examples, install the *Sample Calendar* application. See ["Installing a Productivity and Sample App"](#)

Adding a Calendar to a New Page Using a Local Database

Run the Create Page Wizard to create a new page with a calendar using a local database.

To create a calendar on a new page using a local database:

1. Run the Create Page Wizard:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Create Page**.
The Create Page Wizard appears.
2. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Select a page type - Select **Calendar**.
3. For Page Attributes:

**Tip:**

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#)".

- a. Page Number - If you identify a new page number, the wizard creates a new page. If you identify an existing page number, the wizard adds the component to that page.
 - b. Page Name - Enter a title for the page to be created.
 - c. Page Mode - Select a page mode.
 - d. Page Group - Identify the name of the page group you would like to associate with this page.
 - e. Breadcrumb - Select whether to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - f. Click **Next**.
4. For Navigation Menu, specify the type of navigation to include on this page and click **Next**. The navigation options that display depend upon the current application theme.
5. For Data Source, select **Local Database**.
6. For Source Type, do one of the following:
 - If the source is a table:
 - a. Source Type - Select **Table**.

- b. Table/View Owner - Select the owner of the table on which you are building a calendar.
 - c. Table/View Name - Select the table or view on which the calendar is based.
 - d. Select Columns - Select one or more columns to be included in the calendar.
 - e. Click **Next**.
 - If the Calendar source is a SQL Query:
 - a. Source Type - Select **SQL Query**.
 - b. Enter Region Source - Enter the SQL SELECT statement to be used for the calendar. The SQL SELECT statement must include at least two columns, of which one must be a DATE column.
 - c. Click **Next**.
- 7. For Settings:
 - a. Display Column - Select the column which holds the text displayed for events on this calendar.
 - b. Start Date Column - Select the column which holds the start date for events displayed on this calendar.
 - c. End Date Column - Select the column which holds the end date for events displayed on this calendar. If this attribute is specified, then the calendar displays duration-based events.
 - d. Show Time - Select whether the time portion of the date display.
 - e. Click **Create**.

Adding a Calendar on a New Page Using Remote Database References

Run the Create Page Wizard to add a new page with a calendar that uses a remote database reference.

To create a calendar on a new page:

1. Run the Create Page Wizard:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Create Page**.
The Create Page Wizard appears.
2. For Create a Page:
 - a. (Optional) User Interface - Select a user interface for the page.
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Calendar**.

3. For Page Attributes:

 **Tip:**

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#)".

- a. Page Number - If you identify a new page number, the wizard creates a new page. If you identify an existing page number, the wizard adds the component to that page.
 - b. Page Name - Enter a title for the page to be created.
 - c. Page Mode - Select a page mode.
 - d. Page Group - Identify the name of the page group you would like to associate with this page.
 - e. Breadcrumb - Select whether to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - f. Click **Next**.
4. For Navigation Menu, specify the type of navigation to include on this page and click **Next**. The navigation options that display depend upon the current application theme.
5. For Data Source, , select and then configure a remote database reference:
- For a REST Enabled SQL Service reference:
 - a. Data Source - Select **REST Enabled SQL Service**.
 - b. REST Enabled SQL Service - Select a REST Enabled SQL reference.
 - For a Web source:
 - a. Location - Select **Web Source**.
 - b. Web Source Module - Select a Web Source Module.

 **Tip:**

The UI changes based on your selection. To learn more about an attribute, see field-level Help.

6. For Source Type, do one of the following:
- If the Calendar source is a Table:
 - a. Source Type - Select **Table**.
 - b. Table/View Owner - Select the owner of the table on which you are building a calendar.
 - c. Table/View Name - Select the table or view on which the calendar is based.
 - d. Select Columns - Select one or more columns to be included in the calendar.

- e. Click **Next**.
- If the Calendar source is a SQL Query:
 - a. Source Type - Select **SQL Query**.
 - b. Enter Region Source - Enter the SQL SELECT statement to be used for the calendar. The SQL SELECT statement must include at least two columns, of which one must be a DATE column.
 - c. Click **Next**.
- 7. For Settings:
 - a. Display Column - Select the column which holds the text displayed for events on this calendar.
 - b. Start Date Column - Select the column which holds the start date for events displayed on this calendar.
 - c. End Date Column - Select the column which holds the end date for events displayed on this calendar. If this attribute is specified, then the calendar displays duration-based events.
 - d. Show Time - Select whether the time portion of the date display.
 - e. Click **Create**.



See Also:

- ["Creating a REST Enabled SQL Service Reference"](#)
- ["Managing Web Source Modules"](#)

Creating a Calendar in Page Designer

Manually create a calendar in Page Designer by providing the SQL query.



Tip:

To create a calendar in Page Designer, you must provide a SQL Query. To create a calendar by selecting a table, use the Create Page Wizard. See ["Adding a Calendar to a New Page Using a Local Database"](#).

To create a calendar in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. If necessary, click the Layout tab in the central pane.

3. In the Gallery, click the **Regions** tab.
4. Right-click **Calendar**, click **Add To**, and select the appropriate location.

 **Tip:**

You also select **Calendar** with the mouse and drag it to the appropriate location in the Layout tab.

The Property Editor displays Region attributes. Attributes are organized in groups.

To find a group or attribute:

- **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
- **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

The Messages tab displays a red or yellow badge to identify messages you need to address. Select a message to view the associated attribute in the Property Editor. You must address red error message before you can save.

5. In the Property Editor, edit the appropriate Region attributes:
 - Identification, Title - Enter a region title.
 - Source, Location - Select the location of the database which contains column which holds the CSS Class to style the events displayed on this calendar. Options include:
 - **Local Database** - Data is sourced from the local database.
 - **Remote Database** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL.
 - **Web Source** - Data is sourced from a RESTful web service defined using Web Source Modules.

What displays next depends upon Source, Location you select.

 **Tip:**

To learn more, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. Edit the calendar Attributes.
 - In the Rendering tab, locate the Calendar region and click **Attributes**. The Property Editor displays the calendar Attributes.
 - Edit the calendar Attributes:
 - Settings, Display Column - Select the column which holds the text displayed for events on this calendar.
 - Settings, Start Date - Select the column which holds the start date for events displayed on this calendar.

7. Click **Save**.

Managing Calendar Attributes

Once you create a calendar, you can alter the display by editing attributes in the Property Editor. Each calendar has two sets of editable attributes: region attributes and calendar Attributes.

Region attributes define the area of the page which functions as a container for the calendar. Calendar Attributes specify the template, date columns, and general calendar formatting.

- [Editing Calendar Attributes in the Property Editor](#)
- [Editing an Existing Calendar to Include Add and Edit Functionality](#)
- [Adding Calendar Initialization JavaScript Code](#)
- [Enabling the Dragging and Dropping of Data In an Existing Calendar](#)

Editing Calendar Attributes in the Property Editor

To edit calendar Attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate and select the calendar.
The Property Editor displays the region attributes.
3. In the Rendering tab, locate the calendar and select the **Attributes** node.
The Property Editor displays the calendar attributes. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Edit the calendar attributes.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Editing an Existing Calendar to Include Add and Edit Functionality

Developers can edit an existing calendar to include add and edit functionality.

Tip:

When defining calendar create and edit links for Desktop applications, Oracle recommends defining two separate pages. To create an **Edit Link**, select one of the table or SQL query columns as the Primary Key Column in the Calendar attributes.

To edit an existing calendar to include add and edit functionality:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate and select the calendar.
The Property Editor displays the region attributes.
3. In the Rendering tab, locate the calendar and select the **Attributes** node.
The Property Editor displays the calendar attributes. Attributes are organized in groups.

Tip:

To find a group or attribute:

- **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit **Settings, Create Link** to specify a target page to call when the user clicks an empty cell or an existing calendar entry.
 - a. Settings, Create Link - Click **No Link Defined**.
The Link Builder - Create Link dialog appears.
 - b. Type - Select the type of link target.
 - c. Application - If you select **Page in a different application**, enter the application ID.
 - d. Page - Specify the target page number.

- e. Set Items - Configure the values to be passed from the calendar to the target page. The **Create Link** is called after a date or date range has been selected in the calendar. Therefore, the selected start and end dates must be passed to the Create Page.
 - To pass the selected date or the start date of a selected date range:
 - Name - Specify the name of the target page item containing the start date of the event. If a form, this should be a date picker element.
 - Value - `&APEX$NEW_START_DATE`.
 - When a date range has been selected, pass the end date as follows:
 - Name - Specify the name of the target page item containing the end date of the event. For a form, this should be a date picker element.
 - Value - `&APEX$NEW_END_DATE`.

You can pass additional values in the same way as any other link to an application page.

 - Click **OK**.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Edit **Settings, Edit Link** to define a target page to be called when the user clicks an existing entry.

 **Tip:**

To create an Edit Link, you must pass a primary key value to the target page in order for the form to look up the table row.

- a. Settings, Edit Link - Click **No Link Defined**.
The Link Builder - Create Link dialog appears.
- b. Type - Select the type of link target.
- c. Application - If you select **Page in a different application**, enter the application ID.
- d. Page - Specify the target page number.
- e. Set Items - Configure the values to be passed from the calendar to the target page. The Create Link is called after a date or date range has been selected in the calendar. Therefore, the selected start and end dates must be passed to the Create Page.
 - Name - Specify the name of the target page item containing the primary key value; typically a hidden item, for example:
`PX_ID`.

- Value - Specify the table or SQL query column containing the primary key value, for example:
&ID.
- Click **OK**.

6. Click **Save**.

Adding Calendar Initialization JavaScript Code

You can add a JavaScript function to override the standard jQuery FullCalendar initialization attributes by editing the Initialization JavaScript Code attribute. The function must return a JavaScript Object containing the calendar initialization attributes as parameters.

To add initialization JavaScript code:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In the Rendering tab, under the calendar title, select the **Attributes** node.
The Property Editor displays the calendar attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Advanced**.

Tip:

To search for a group or an attribute, enter a keyword in the Filter Properties field. The groups and attributes containing the term appear.

5. For Initialization JavaScript Code, enter a JavaScript function to override the standard jQuery FullCalendar initialization attributes.
For example:

```
function ( pOptions) {
    pOptions.titleFormat      = "[Conference
Schedule]";                 // custom title
    pOptions.minTime          =
"07:00:00";                 // hide slots before minTime
    pOptions.maxTime          =
"18:00:00";                 // hide slots after maxTime
    pOptions.columnFormat    = {month: '', week: 'dddd', day:
```



```
'dddd'}; // week view column headings
    pOptions.slotDuration =
"00:15:00"; // custom slot duration
    pOptions.weekNumbers =
true; // show week numbers
    pOptions.weekNumberTitle =
"CW";
    return pOptions;
}
```

6. Click **Save**.



Note:

To view examples of this functionality, run the *Sample Calendar* sample application. From the Navigation menu, select **Custom Calendar Initialization**. See "[Installing a Productivity and Sample App](#)."

Enabling the Dragging and Dropping of Data In an Existing Calendar

The Calendar Region supports moving or changing calendar events with drag and drop.

To enable support for drag and drop for a calendar region:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In the Rendering tab under the calendar title, select the **Attributes** node.

The Property Editor displays the attributes for the calendar. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Settings** and edit the following attributes:
 - a. Settings, Primary Key Column - Select a table or SQL Query column.

The Drag and Drop field appears in the Property Editor, below Additional Calendar Views.
 - b. Settings, Drag and Drop - Select **Yes**.

Drag and Drop PL/SQL Code appears.

- c. Settings, Drag and Drop PL/SQL Code - Enter the PL/SQL code to execute when the end user finishes a drag and drop for an event.

Within the PL/SQL code, you can reference the ID of the dragged event, the new start, and end end dates as:

```
:APEX$PK_VALUE, :APEX$NEW_START_DATE amd :APEX$_NEW_END_DATE
```

For example, the following code changes the event row in the `PROJECTS` table after the end user has dragged the event to a new date.

```
begin
  update projects
    set start_date = to_date(:APEX$NEW_START_DATE,
'YYYYMMDDHH24MISS'),
        end_date = to_date(:APEX$NEW_END_DATE,
'YYYYMMDDHH24MISS')
    where id = :APEX$PK_VALUE;
end;
```

To learn more, click the **Help** tab in the central pane.

5. Click **Save**.

About Dynamic Action Support for Calendar

Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. Using the Dynamic Action Create wizard, you specify an action that is performed when a defined set of conditions occur. You can also specify which elements are affected by the action, and when and how they are affected.

When working with dynamic actions, you should be mindful of the fact that the more dynamic actions you add to a page, the greater your overall page size. This is because the dynamic action framework emits additional code to the client for each dynamic action defined, which then also must be downloaded and executed by the framework in the client.

Dynamic actions contain a condition that can be specified to control the action. Oracle Application Express components such as reports or charts can react to specific calendar events. For example, if you view a calendar and click a button to view a subsequent month, you can create a dynamic action to automatically refresh an associated report.

To use this feature the developer creates a new dynamic action on the Calendar page in Page Designer. Select the new dynamic action in the Dynamic Action tab in the left pane of Page Designer. Once created, the new dynamic action displays in red with the label **New**. Then, the developer edits the dynamic action attributes in the Property Editor. In Property Editor, locate and expand **When** attribute. Under the **When** attribute, for **Event**, select one of the following events:

- **Date Selected [Calendar]**

This event fires when the user selects an empty date or date range either with the mouse or with the keyboard. If the developer provides a "create link" in the calendar attributes, no event fires.

Information about the current view is being passed as the "data" object:

```
{ "newStartDate"::: "newEndDate"::: }  
  
{ "newStartDate":{first day of selection} (YYYYMMDDHH24MISS),  
  "newEndDate": {last day of selection} YYYYMMDDHH24MISS }
```

- **Event Selected [Calendar]**

This event fires when the user clicks an existing event with the mouse or selects it with the keyboard. When the developer provided an "edit link" in the Calendar attributes, no event is being fired. The "data" object contains the jQuery FullCalendar JSON representation of the event.

 **See Also:**

Full Calendar documentation Event Data, "[Event Object](#)"

- **View Changed [Calendar]**

This event fires when either the view type changes (day, week, month, list) or when the view changes to its previous or next view page. The "data" object contains the following information:

```
{ "viewType": {view type, day, week, month or list},  
  "startDate": {first day of view} (YYYYMMDDHH24MISS),  
  "endDate": {last day of view} (YYYYMMDDHH24MISS) }
```

 **Note:**

To view examples of this functionality, run the Sample Calendar sample application. From the navigation menu, select the **Calendar and Dynamic Actions** and select the calendar event. See to "[Installing a Productivity and Sample App.](#)"

 **See Also:**

"[Managing Dynamic Actions](#)" and "[Debugging Dynamic Actions](#)" for information on how to debug problems

Creating Maps

App Builder includes built-in wizards for generating map charts on pages with a Desktop user interface. How you create a map chart depends upon whether you are adding the map to an existing page, or adding a map on a new page.

- [About Map Chart Support](#)
Map chart support in Oracle Application Express is based on the AnyChart AnyMap Interactive Maps Component.
- [About Creating SQL Queries for Maps](#)
You define a map chart by providing a SQL query with a specific syntax.
- [Adding a Map Chart to a New Page](#)
Add a map chart to a page by running the Create Page Wizard.
- [Adding a Map Chart in Page Designer](#)
Add a map chart by manually creating it in Page Designer.
- [Managing Map Charts](#)
Once you create a map chart, you can alter its display by editing attributes in the Property Editor.

About Map Chart Support

Map chart support in Oracle Application Express is based on the AnyChart AnyMap Interactive Maps Component.

AnyMap is a flexible Macromedia Flash-based solution that enables developers to visualize geographical related data. Map charts are rendered by a browser and require Flash Player 9 or later. For more information about AnyChart, go to <http://www.anychart.com>

AnyChart stores map data in files with a *.amap extension, and supports 300 map files for the United States of America, Europe, Asia, Europe, Africa, Oceania, North America, and South America. To render a desired map, you select the map source in the wizard (for example, Germany) and the map XML automatically references the desired map source .amap file, germany.amap. To browse through the available maps, see "Map Reference" in *AnyChart User's Guide*: <http://6.anychart.com/products/anymap/docs/>

About Creating SQL Queries for Maps

You define a map chart by providing a SQL query with a specific syntax.

When you define a map chart providing a SQL query using the following syntax:

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

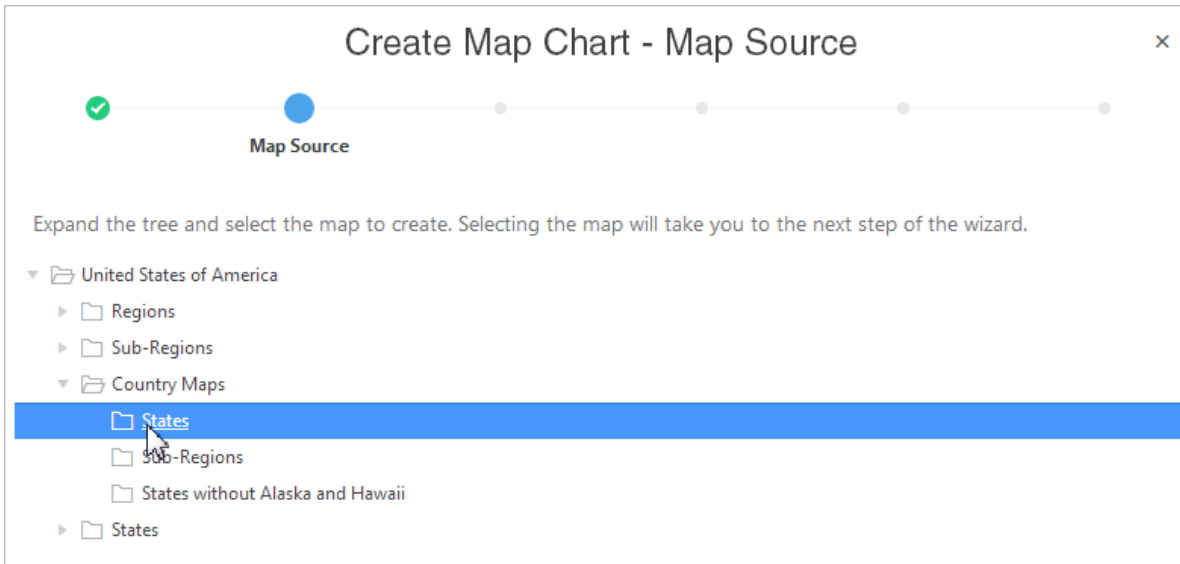
Where:

- *link* is a URL.
- *label* is the text that identifies the point on the map with which you want to associate data. The Region ID or Region Name of the map will be used as the label.
- *value* is the numeric column that defines the data to be associated with a point on the map.

Consider the following example when creating a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

3. Click **Create Page**.
4. Click **Chart**.
5. Select **Map Chart**.
6. For Map Type, select **United States of America** and click **Next**.
7. For Map Source, expand Country Maps and select **States**.



8. For Page Attributes specify the appropriate attributes and click **Next**.
9. For Navigation Preference, specify the type of navigation to include on this page and click **Next**. The navigation options (for example, navigation menu or tabs) depends upon the current application theme.
10. For Map Attributes, select the appropriate attributes and click **Next**.
The Query page appears.
11. Scroll down and expand and review the Map Query Example and Map Reference Information regions.
12. Scroll down and expand and review the **Map Reference Information** region at the bottom of the page.

Create Map Chart

Query

✓
✓
✓
✓
✓
●

Enter SQL Query or PL/SQL function returning a SQL Query:

```
FROM (
SELECT 'Ireland' country, 4203200 people
FROM dual)
```

Build Query

Perform query validation Save query without validation ?

Page Items to Submit: ^ ?

Maximum Rows: ?

When No Data Found Message: ?

> Map Query Example

▼ Map Reference Information

| Region Id | Region Name | Centroid X | Centroid Y | State Fips | Sub Region | Gmi Admin |
|-----------|-------------|-------------------|-----------------|------------|------------|-----------|
| AK | Alaska | -151.777777777778 | 65.676268861454 | 2 | Pacific | |
| ... | ... | ... | ... | ... | ... | ... |

< Cancel
Create

Specify a SQL query by either:

- Entering a SQL query in the field provided. See "[About Creating SQL Queries for Maps.](#)"
- Clicking the **Build Query** button. When the Build Chart Query Wizard appears, follow the on-screen instructions.

When you create your query, you must incorporate the region name information as described in the Map Reference Information region. The following example includes information for Florida and Alaska:

```
SELECT null link, region label, value Populations
FROM (
SELECT 'Florida' region, 18328340 value FROM dual
UNION ALL
SELECT 'Alaska' region, 686293 value FROM dual)
```

Note:


If you are referencing geographical information stored in a database table, the information must correspond with the Region information associated with the map source. For more information, see *AnyChart User's Guide*: <http://6.anychart.com/products/anymap/docs/>

Click **Create**.

Adding a Map Chart to a New Page

Add a map chart to a page by running the Create Page Wizard.

To add a map chart on a new page:

1. On the Workspace home page, click the **App Builder** icon.
 2. Select an application.
 3. Click **Create Page**.
The Create Page Wizard appears.
 4. For Create a Page:
 - a. User Interface - Select a user interface for the page.
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Select a page type - Select **Chart**.
 5. On Create Chart, select **Map Chart**.
 6. On Create Page, select a map type (for example, **Europe**), and click **Next**.
 7. On Create Map Chart, expand the tree and select the map to create.
 8. For Page Attributes:
 - a. Page Number - Select a page in which the chart object is to appear.
 - b. Page Name - If creating a new page, enter a page name.
 - c. Page Mode - Select a page mode.
 - d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use
 - e. Click **Next**.
 9. On Navigation Menu, specify a Navigation Preference to include on this page and click **Next**. The navigation options (for example, navigation menu or tabs) depends upon the current application theme.
 10. For Map Attributes, select the appropriate attributes and click **Next**.
-  **Note:**
To learn more about an attribute, see field-level Help.
11. Specify a query by either:
 - Entering a SQL query in the field provided.
 - Clicking the **Build Query** button. When the Build Chart Query Wizard appears, follow the on-screen instructions.

Specify relevant attributes. To view map reference information associated with your selected Map Source, expand the **Map Reference Information** region at the bottom of the page.

To associate data with geographical locations, you must incorporate information from the Region Name or Region ID columns into the Map query.

12. Click **Create**.



See Also:

["About Creating SQL Queries for Maps"](#)

Adding a Map Chart in Page Designer

Add a map chart by manually creating it in Page Designer.

To add a map chart to an existing page.

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. From the Gallery select the **Map Chart** region and drag it to the appropriate location in the Layout tab.



Tip:

You can also right-click **Map Chart** region, select **Add To**, and select the appropriate location.

Page Designer indicates what actions are required next.

The Messages tab displays a red or yellow badge indicating messages you need to address. The Message tab displays two types of messages:

- **Errors** - Error messages display in red. Selecting an error message displays the associated attribute in red in the Property Editor. You must address errors before a page can be saved.
 - **Warnings** - Warning messages display in yellow. Selecting a warning message displays the associated attribute in yellow in the Property Editor. You must address errors before a page can be saved. Note you can save a page without addressing warning messages.
3. In the Property Editor, edit the following attributes.



Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

- a. Title - Enter a title for the region.
 - b. Layout:
 - Sequence - Enter the display sequence for this item.
 - Parent Region - Select the parent region to which this region belongs.
 - Position - Select the template position used to display this region.
 - c. Appearance:
 - Template - Select a region template.
 - Item Display Position - Specify where page items display in relation to the main region content.
 - d. Layout - Specify the Sequence, Parent Region, and Position.
 - e. Appearance - Specify the template.
4. View the Map Chart attributes. In the Rendering tab, under the map chart title, select the **Attributes** node.

In the Property Editor, edit the following Map attributes:

- a. Map, Level 1 - Select the map level from the available list (for example **Europe**).
- b. Map, Level 2 - Select the map level from the available list (for example, **Denmark**).
- c. Source, SQL Query - Enter the SQL Query for the chart series.
- d. Edit other attributes as appropriate.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Click **Save**.

Managing Map Charts

Once you create a map chart, you can alter its display by editing attributes in the Property Editor.

- [Editing Map Chart Attributes](#)
- [Using Custom XML with Map Charts](#)
- [Enabling Automatic Updates](#)

Editing Map Chart Attributes

To view the map chart attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Select a page.
Page Designer appears.
2. To edit region attributes:
 - a. In the Rendering tab, select the region that containing the map chart.
The region attributes display in the Property Editor.
 - b. Edit the appropriate attributes.
3. To edit chart map attributes:
 - a. In the Rendering tab, under the region title, select the **Attributes** node.
The map chart attributes display in the Property Editor.
 - b. Edit the appropriate attributes.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

4. Click **Save**.

 **See Also:**

["Using Custom XML with Map Charts"](#) and ["Enabling Automatic Updates"](#)

Using Custom XML with Map Charts

There are additional map settings that cannot be controlled using the standard attributes on the Map Attributes page. To further control the look and feel of a chart, you can use custom XML.

To use custom XML:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
Page Designer appears.
2. In the Rendering tab, under the region title, select the **Attributes** node.
The map chart attributes display in the Property Editor.
3. Locate and expand the **Custom XML**.
4. For Custom XML, Custom, select **Yes**.
5. Edit the XML.
6. Click **Save**.

**Tip:**

For more information on supported XML format for maps, see the Online Map XML Reference at:

<http://6.anychart.com/products/anymap/docs/>

Enabling Automatic Updates

You can create map charts that monitor information by enabling the Asynchronous Update attribute on the Map Attributes page. Enabling this attribute updates the map to reflect changes in the underlying data within a specified time interval.

To enable automatic refresh:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, select the region that contains the map chart.
The region attributes display in the Property Editor.
3. In the Rendering tab, under the region title, select the **Attributes** node.
The map chart attributes display in the Property Editor.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Find **Automatic Refresh**.
 - a. Automatic Refresh - Select **Yes**.
 - b. Interval - Enter the interval in seconds between chart updates.

**Note:**

Oracle discourages very small updates intervals, such as 2 seconds, since they may cause serious database performance issues.

6. Click **Save**.

Creating Charts

You can add a chart by running a built-in wizard or creating it manually in Page Designer.

- [About Supported Charts](#)
App Builder supports two types of charts: Charts (based on the Oracle JavaScript Extension Toolkit (Oracle JET) Data Visualizations) and AnyChart Charts.
- [Sample Charts](#)
Developers can view chart examples by installing the *Sample Charts* sample application.
- [Adding a Chart by Running the Create Page Wizard](#)
Add a chart to a new page by running the Create Page Wizard.
- [Adding a Chart on a New Page Using Remote Database References](#)
Run the Create Page Wizard to add a new page with a chart using a remote database reference.
- [Adding a Chart in Page Designer](#)
Add a chart by creating it manually in Page Designer.
- [Managing Charts](#)
You can alter how a chart displays by editing chart attributes in Page Designer.
- [Using Custom JavaScript with Charts](#)
Control the look and feel of a chart by adding custom JavaScript.



See Also:

["Managing Legacy Charts"](#)

About Supported Charts

App Builder supports two types of charts: Charts (based on the Oracle JavaScript Extension Toolkit (Oracle JET) Data Visualizations) and AnyChart Charts.

App Builder supports the following charts:

- **Chart** - Chart support in Oracle Application Express is based on the Oracle JET Data Visualizations. Oracle JET empowers developers by providing a modular open source toolkit based on modern JavaScript, CSS3, and HTML5 design and development principles. 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. Each Oracle JET visualization supports animation, accessibility, responsive layout, internationalization, test automation, and a range of inter activity features. The charts provide dozens of different ways to visualize a data set, including bar, line, area, range, combination, scatter, bubble, polar, radar, pie, donut, funnel, and stock charts..

 **See Also:**

"Working with Data Visualizations" in *JavaScript Extension Toolkit (JET) Developing Applications with Oracle JET*.

- **AnyChart Chart** - AnyChart chart support is based on a third party charting solution provided by AnyChart. This is a flexible Flash and JavaScript (HTML5) based solution that enables developers to create animated and compact interactive charts. AnyChart charts have been categorized as a legacy component and may be deprecated in the future release.

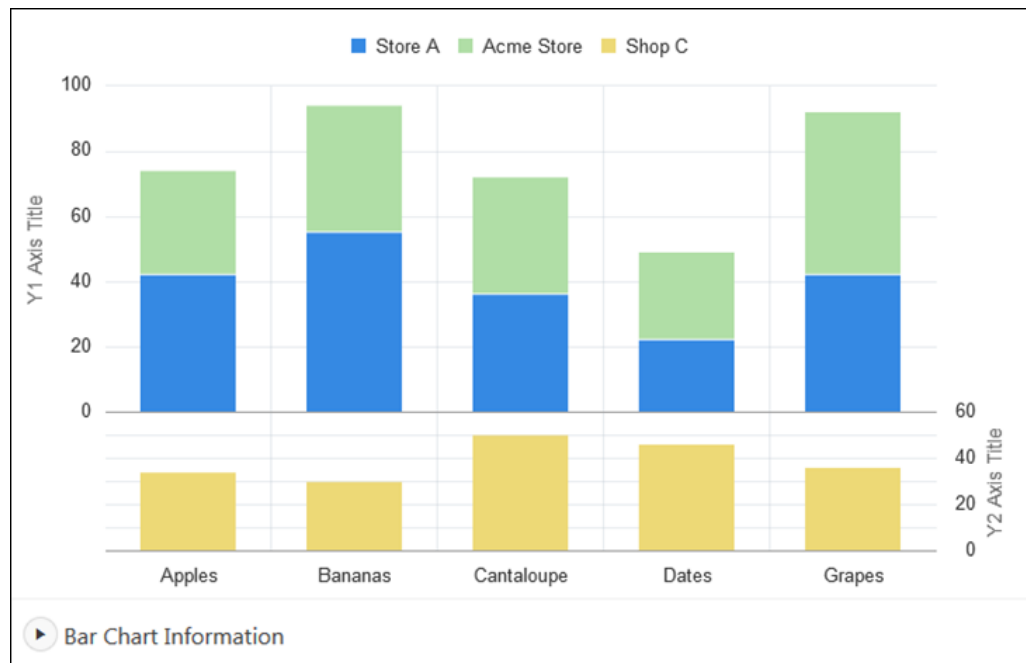
 **See Also:**

"Managing Legacy Charts" and "Upgrading an Application to Include New Components"

Sample Charts

Developers can view chart examples by installing the *Sample Charts* sample application.

The following is a sample chart from *Sample Charts* sample application.



The chart includes the following attributes:

- Extra Y axis - Displays the title *Y2 Axis Title*, and associated with *Shop C* series of data.

- Split Dual Y Axis - Displays two series of data, *Store A* and *Acme Store*, stacked and associated with the Y axis, and the third series of data, *Shop C*, associated with an extra Y axis and rendered as a separate chart below the other two series of data.

Expand the Bar Chart Information region at the bottom of the page to view information on the chart.

 **See Also:**

"[Installing a Productivity and Sample App](#)" for more information on installing the *Sample Charts* application

Adding a Chart by Running the Create Page Wizard

Add a chart to a new page by running the Create Page Wizard.

To add a chart on a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

The Create Page Wizard appears.

4. For Create a Page:
 - a. (Optional) User Interface - Select a user interface for the page.
 - b. Select a page type - Select **Chart**.
5. For Chart Type, select a chart type.

 **Note:**

If you select **Gantt**, you are prompted to select a Gantt chart type.

6. For Page and Region Attributes:
 - a. Page Number - Enter a page in which the chart object is to appear.
 - b. Page Name - Enter a page name.
 - c. Page Mode - Select a page mode.
 - **Normal** - The page displays as a normal Application Express application page.
 - **Modal Dialog** - The page displays as a modal dialog. A modal dialog is an overlay window which remains active and focused until the end user closes it. The underlying page is grayed out which prevents the end user from interacting with the rest of the page until the dialog closes.
 - d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page.

- e. Click **Next**.
7. For Navigation Preference, specify the type of navigation to include on this page and click **Next**. The navigation options (for example, navigation menu or tabs) depends upon the current application theme.

 **Tip:**

The steps that follow assume you selected an Oracle JET chart. If you select a non-Oracle JET chart (for example, Gantt) the steps that follow may differ. Follow the on-screen instructions.

8. For Source, do one of the following:
 - If the Chart source is a Table:
 - a. Source Type - Select **Table**.
 - b. Table/View Owner - Select the owner of the table on which you are building the chart.
 - c. Table/View Name - Select the table or view on which the chart is based.
 - d. Page Items to Submit - Enter a comma separated list of page items on the current page to be set into session state when the chart data gets read with a separate request.
 - e. Maximum Rows - Enter the maximum number of rows you want to use to display the chart.
 - f. Click **Next**.
 - If the Chart source is a SQL Query:
 - a. Source Type - Select **SQL Query**.
 - b. SQL Query - Enter the `SQL SELECT` statement to be used for the chart. The `SQL SELECT` statement must include at least two columns.
 - c. Page Items to Submit - Enter a comma separated list of page items on the current page to be set into session state when the chart data gets read with a separate request.
 - d. Maximum Rows - Enter the maximum number of rows you want to use to display the chart.
 - e. Click **Next**.
9. For Column Mapping:
 - a. Select the columns to be mapped to the chart. The options that display vary depending upon the chart type selected. To learn more about an attribute, click the field-level Help.
 - b. Click **Next**.
10. Click **Create**.

Adding a Chart on a New Page Using Remote Database References

Run the Create Page Wizard to add a new page with a chart using a remote database reference.

To add a chart on a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

The Create Page Wizard appears.

4. For Create a Page:
 - a. (Optional) User Interface - Select a user interface for the page.
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Select a page type - Select **Chart**.
5. For Chart Type, select a chart type.

 **Note:**

If you select **Gantt**, you are prompted to select a Gantt chart type.

6. For Page and Region Attributes:
 - a. Page Number - Enter a page in which the chart object is to appear.
 - b. Page Name - Enter a page name.
 - c. Page Mode - Select a page mode.
 - **Normal** - The page displays as a normal Application Express application page.
 - **Modal Dialog** - The page displays as a modal dialog. A modal dialog is an overlay window which remains active and focused until the end user closes it. The underlying page is grayed out which prevents the end user from interacting with the rest of the page until the dialog closes.
 - d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page.
 - e. Click **Next**.
7. For Navigation Preference, specify the type of navigation to include on this page and click **Next**. The navigation options (for example, navigation menu or tabs) depends upon the current application theme.

 **Tip:**

The steps that follow assume you selected an Oracle JET chart. If you select a non-Oracle JET chart (for example, Gantt) the steps that follow may differ. Follow the on-screen instructions.

8. For Source, select and then configure a remote database reference:
 - For REST Enabled SQL Service:
 - a. Location - Select **REST Enabled SQL Service**.

- b. REST Enabled SQL Service - Select a REST Enabled SQL reference.
- For Web source:
 - a. Location - Select **Web Source**.
 - b. Web Source Module - Select a Web Source Module.
- 9. For Source Type, do one of the following:
 - If the Chart source is a Table:
 - a. Source Type - Select **Table**.
 - b. Table/View Owner - Select the owner of the table on which you are building the chart.
 - c. Table/View Name - Select the table or view on which the chart is based.
 - d. Page Items to Submit - Enter a comma separated list of page items on the current page to be set into session state when the chart data gets read with a separate request.
 - e. Maximum Rows - Enter the maximum number of rows you want to use to display the chart.
 - f. Click **Next**.
 - If the Chart source is a SQL Query:
 - a. Source Type - Select **SQL Query**.
 - b. SQL Query - Enter the `SQL SELECT` statement to be used for the chart. The `SQL SELECT` statement must include at least two columns.
 - c. Page Items to Submit - Enter a comma separated list of page items on the current page to be set into session state when the chart data gets read with a separate request.
 - d. Maximum Rows - Enter the maximum number of rows you want to use to display the chart.
 - e. Click **Next**.
- 10. For Column Mapping:
 - a. Select the columns to be mapped to the chart. The options that display vary depending upon the chart type selected. To learn more about an attribute, click the field-level Help.
 - b. Click **Next**.
- 11. Click **Create**.



See Also:

- ["Creating a REST Enabled SQL Service Reference"](#)
- ["Managing Web Source Modules"](#)

Adding a Chart in Page Designer

Add a chart by creating it manually in Page Designer.

To add a chart to an existing page in Page Designer:

1. View the page to contain the chart in Page Designer.
2. In the Gallery at the bottom of the central pane, click **Regions** and locate Chart. The Gallery lists all controls or components you can add to a page. Passing the cursor over a control or component displays a tooltip that describes it.
3. From the Gallery, right-click **Chart** to view a context menu. Select **Add To** and then the desired location.

 **Tip:**

You can also click **Chart** in the Gallery and drag it to the appropriate location in the Layout tab.

Page Designer indicates what actions are required next.

4. (Optional) Click the **Messages** tab in the central pane.

The Messages tab displays a red or yellow badge indicating messages you need to address. The Message tab displays two types of messages:


- **Errors** - Error messages display in red. Selecting an error message displays the associated attribute in red in the Property Editor. You must address errors before a page can be saved.
 - **Warnings** - Warning messages display in yellow. Selecting a warning message displays the associated attribute in yellow in the Property Editor. You must address errors before a page can be saved. Note you can save a page without addressing warning messages.
5. Do one of the following to locate the required fields highlighted in the Property Editor groups:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click the **Go to Group** icon and select the group. To return Property Editor to the default display, click the **Go to Group** icon again and select **Expand All**.
 6. Edit the required region attributes.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. (Optional) Select the location of the database you want to use. Options include:

- **Local Database** - Data is sourced from a local database. (This location is selected by default.)
 - **Remote Database** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL Reference.
 - **Web Source** - Data is sourced from a RESTful web service defined using Web Source Modules.
8. Edit the chart attributes:
 - a. In the Rendering tab, select the **Attributes** node.
The Property Editor displays Attributes.
 - b. Edit the required chart attributes.
 9. Edit the series:
 - a. In the Rendering tab, select the **Series** node.
The Property Editor displays Series.
 - b. Enter the series SQL query and define its associated column mappings for the chosen chart type.
 10. Edit the axes:

 **Tip:**

The Axes node only displays for chart types that support axes (such as area, bar, bubble, combination, line, line with area, polar, radar, range, scatter, and stock).

 - a. In the Rendering tab, select the **Axis** node.
The Property Editor displays Axis.
 - b. Edit the required axis attributes and define any data formatting to be applied to the axis values.
 11. Click **Save** or **Save and Run Page**.

Managing Charts

You can alter how a chart displays by editing chart attributes in Page Designer.

- [Editing Chart Attributes](#)
- [Switching Chart Type](#)
- [Enabling Automatic Refresh](#)



See Also:

"Viewing a Page in Page Designer"

Editing Chart Attributes

To edit chart attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the left pane, locate the chart in the Rendering tab.
3. Under the chart, select the **Attributes** node.
Property Editor displays chart Attributes. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Edit the chart attributes.
6. Click **Save**.

Switching Chart Type

Once you create a chart, you can switch its chart type by editing chart attributes.

To switch a chart type:

1. View the page to contain the chart in Page Designer.
2. In the Rendering tab, select the **Attributes** node.
Property Editor - Attributes appears.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Chart** section.
5. For Type, select the chart type you want to switch to. For example, Combination.
To learn more about an attribute, see field-level Help.
6. In the Rendering tab, select the **Series** node
Property Editor - Series appears.
 - a. Ensure all required Column Mapping columns have been defined.

 **Note:**

Column mappings may differ for different chart types, so you must ensure all required column mappings are defined for the selected chart type.

- b. For Type, select the chart series type. For example, Bar Range.

 **Note:**

Only certain chart types support combining different types of series on the same data plot. The Series Type attribute will only be visible for chart types combination, polar, radar, and range.

7. Click **Save** or **Save and Run Page**.

Enabling Automatic Refresh

Charts can monitor information by enabling the Automatic Refresh attribute on the Chart attributes page, or using a dynamic action with the Refresh action.

- [Enabling the Automatic Refresh Attribute](#)
- [Creating a Refresh Dynamic Action](#)

Enabling the Automatic Refresh Attribute

Enabling the Automatic Refresh attribute updates the chart to reflect changes in the underlying data within a specified time interval.

To enable automatic refresh updates:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region containing the chart and select the **Attributes** node
The attributes display in the Property Editor.
3. Find **Automatic Refresh**.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.

- **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. For Automatic Refresh, select **Yes**.
 6. Click **Save** .

Creating a Refresh Dynamic Action

To create a Refresh dynamic action:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane.
3. Under Dynamic Actions, right-click **Events** and select **Create Dynamic Action**.
Attributes for the dynamic action display in the Property Editor.

Tip:

The Messages tab displays a red or yellow badge to identify messages you need to address. Selecting a message displays the associated attribute in the Property Editor. You must address red error message before you can save.

4. In the Property Editor, edit the following Dynamic Action attributes:

Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

- a. Identification, Name - Enter a name of the dynamic action.
 - b. Execution Options, Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - c. When, Event - Specify the event that causes the dynamic action to fire.
 - d. When, Selection Type - Select the type of page element or construct to be used to trigger the event. For examples, select the attribute in the Property Editor and click the Help tab in the center pane.
5. In the Rendering tab, select the first action under True:
 - a. Action - Select **Refresh**.
 - b. Selection Type - Select **Region**.
 - c. Region - Select the region containing the chart.

6. Click **Save**.

**See Also:**

["Managing Dynamic Actions"](#)

Using Custom JavaScript with Charts

Control the look and feel of a chart by adding custom JavaScript.

To use custom JavaScript:

**Tip:**

The *Sample Charts* application contains a number of examples of using custom JavaScript with chart. See ["Installing a Productivity and Sample App"](#) for more information on installing sample applications.

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region containing the chart and select the **Attributes** node.

The Property Editor displays attributes for the page. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Advanced**.
5. In JavaScript Code, enter the code to customise your chart.

For example:

```
function( options ){  
    // Setup a callback function which gets called  
    // when data is retrieved, it allows to manipulate the series  
    options.dataFilter = function( data ) {  
  
        // e.g Set the first series of chart to the colour red  
        data.series[ 0 ].color = "red";  
    }  
}
```

```
};  
// Set chart initialization options  
// e.g. Set chart type to Line  
options.type = "line";  
return options;  
}
```

6. Click **Save**.

 **Tip:**

For more information on supported chart options, see "[ojChart](#)" in JSDoc pages for the Oracle JavaScript Extension Toolkit (JET).

 **See Also:**

["Adding a Legacy Chart to a New Page"](#)

Creating Help for Your Application

Help created in App Builder displays on a dedicated Help page. To access Help, users click a link that takes them to a dedicated Help page. This Help page displays page and field-level Help topics specific to the page they are viewing.

- [Creating Help](#)
Developers can create both a dedicated Help page and item Help text for an application.
- [Using the Bulk Edit Item Help Report](#)
Edit all Help topics within an application at once using the Bulk Edit Item Help report
- [Seeding Item Help Topics](#)
For applications that do not yet contain Help, you can perform a mass update (or seed) of default Help text.
- [Creating a Help Navigation Bar Entry](#)
Once you have created your Help, the next step is to create a navigation bar entry so users can link to it.

Creating Help

Developers can create both a dedicated Help page and item Help text for an application.

- [About Creating Help](#)
- [Creating a Help Page and Region](#)
- [Defining Page Help](#)
- [Defining Item Help](#)

About Creating Help

App Builder includes built-in attributes to create Help for your application. Creating Help for your application involves the following steps:

- **Step 1:** Create a dedicated Help page and Help region. See "[Creating a Help Page and Region](#) ."
- **Step 2:** Define page Help text. See "[Defining Page Help](#) ."
- **Step 3:** Define item Help text. See "[Defining Item Help](#) ."
- **Step 4:** Create a navigation bar icon to link to your Help page. See "[Creating a Help Navigation Bar Entry](#)."

Creating a Help Page and Region

To create a Help page and region in Page Designer:

1. Create a blank page for your Help.
2. View the new page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

3. In the Gallery, right-click **Help Text** region, select **Add To**, and select the appropriate location.

Tip:

You can also drag and drop from the Gallery. From the Gallery select the **Help Text** region and drag it to the appropriate location in the Layout tab.

4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Edit the following attributes:
 - a. Identification, Title - Enter a title for the region.
 - b. Layout, Sequence - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.
 - c. Layout, Parent Region - Select the parent region to which this region belongs. If a parent region is selected then this region is rendered completely inside the parent region.

- d. Layout, Position - Specify the Sequence, Parent Region, and Position.
- e. Appearance, Template - Select a region template to define the appearance and layout of this region. Region templates are defined in the application theme. When a region is created, the template is automatically set to the default region template defined in the current theme.
- f. Appearance, Item Display Position - Select where page items display in relation to the main region content.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

6. Click **Save**.

 **Tip:**

For more information on supported chart options, see "[ojChart](#)" in JSDoc pages for the Oracle JavaScript Extension Toolkit (JET).

7. Click **Save**.

 **See Also:**

["Adding a New Page to an Application"](#)

Defining Page Help

To define Page Help text:

1. View the new page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, click the page name.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. In Property Editor - Page, find **Help** and enter text in **Help Text** attribute.

5. Click **Save**.

Defining Item Help

To define item Help:

1. View the new page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, click item for which you want to define Help.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. In the Property Editor, find **Help** and enter text in **Help** attribute.
5. Click **Save**.
6. Repeat the previous steps for each item requiring Help text.

Using the Bulk Edit Item Help Report

Edit all Help topics within an application at once using the Bulk Edit Item Help report

Developers can use the Bulk Edit Item Help report to view the item Label and Name, access the item, or link to the page containing the item.to the associate page

If you are including Help in your application, you can edit multiple Help topics at once using the Bulk Edit Item Help report.

To view the Bulk Edit Item Help report:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, select **Item Utilities**.
5. Click **Grid Edit of All Item Help Text**.
6. In Bulk Item Help Report, you can:
 - Update existing Help topics. Edit the Help text that appears and click **Apply Changes**.
 - Link to the page containing the item by clicking the page number.
 - Link to the Page Item by clicking the item name.



See Also:

"Seeding Item Help Topics"

Seeding Item Help Topics

For applications that do not yet contain Help, you can perform a mass update (or seed) of default Help text.

To seed Help topics:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, select **Item Utilities**.
5. Click **Grid Edit of All Item Help Text**.
6. Click **Seed Item Help Text**.
7. To edit existing item Help:
 - a. In Item Help Text, enter the default text to appear in all Help topics.
 - b. Click **Apply Changes**.
8. To create help text for all items that currently do not have help text, click **Seed Item Help Text** and follow the on-screen instructions.
9. Click **Apply Changes**.

Creating a Help Navigation Bar Entry

Once you have created your Help, the next step is to create a navigation bar entry so users can link to it.

To create a navigation bar entry:

1. Navigate to the Navigation Bar Entries page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click **Create**.

The Create Navigation Bar Entry Wizard appears.
3. For Method, select **From Scratch** and click **Next**.
4. For Type, select **Navigation to URL** and click **Next**.
5. For Attributes, specify the following:

- a. Sequence - Specify the order of evaluation for this component.
 - b. Entry Label - Enter display text for this navigation bar entry.
 - c. Icon Image Name - Enter an image name. For naming conventions, see field-level Help.
 - d. Image ALT - Enter ALT text for navigation icons that are images. If you do not specify an image name, then this text displays.
 - e. Image Height - Defines the height of the image.
 - f. Width - Defines the width of the image.
 - g. Click **Next**.
6. For Target:
- a. Target is a - Select **Page in this application**.
 - b. Page - Specify the page number of the help page you created in "[Creating Help](#)."
 - c. Request - Enter the following:

`&APP_PAGE_ID.`

By specifying substitution string `&APP_PAGE_ID` as the Request, you are instructing the Application Express engine to display Help text for the current page when the user clicks this icon.
 - d. Click **Next**.
7. To set a condition for displaying the navigation bar entry, select a condition type.
8. Click **Create**.

12

Using Themes and Theme Styles

Developers can alter a database application's user interface and page layout through themes and templates.

- [Using Themes](#)
Use themes and theme styles to increase your productivity when building an application.
- [Managing Themes and Subscriptions](#)
Developers can manage themes directly or through the use of theme subscriptions.
- [Creating Custom Themes](#)
Create a custom theme by modifying an existing templates.
- [Using Custom Cascading Style Sheets](#)
Upload or reference a cascading style sheet.

Using Themes

Use themes and theme styles to increase your productivity when building an application.

- [About Themes](#)
Themes are collections of templates that enable developers to define the layout and style of an entire application.
- [About Responsive Design and the Universal Theme](#)
Universal Theme - 42 (Universal Theme) enables developers to build modern web applications without requiring extensive knowledge of HTML, CSS, or JavaScript
- [Using Theme Styles and Theme Roller](#)
A theme style is a CSS that is added to the base CSS. Using the Theme Roller utility, developers can change the appearance of an application. Universal Theme - 42 includes theme styles.
- [Understanding Template Options](#)
Template options provide developers with a declarative approach to applying different styles to components on an Oracle Application Express page.
- [Accessing the Themes Page](#)
Manage themes on the Themes page.
- [Editing Themes](#)
Edit a theme by editing theme attributes.

**See Also:**

["Managing Themes and Subscriptions"](#) and ["Managing Themes and Subscriptions"](#)

About Themes

Themes are collections of templates that enable developers to define the layout and style of an entire application.

Themes provide developers with a complete set of templates that accommodate every UI pattern that may be needed in an application. Templates are organized first by template type and then by template class. Template types include page, region, report, list, button, label, and popup list of values (LOV).

Each template type has a number of template classes. A template class defines the purpose of a template within a template type. For example, a region template can be classified as a form region template, a report region template, and so on. These classifications enable Oracle Application Express to map templates among themes, making it easy to quickly change the entire look and feel of an application.

Administrators can add themes to the theme repository as follows:

- **Workspace Themes** - Workspace administrators can create themes that are available to all developers within the workspace. See "[Managing Workspace Themes](#)."
- **Public Themes** - Instance administrators can create public themes by adding them to the Oracle Application Express Administration Services. Once added, these public themes are available to all developers across all workspaces in an instance. See "Managing Public Themes" in *Oracle Application Express Administration Guide*.

About Responsive Design and the Universal Theme

Universal Theme - 42 (Universal Theme) enables developers to build modern web applications without requiring extensive knowledge of HTML, CSS, or JavaScript

About Responsive Design

Responsive design enables you to design web pages so that the layout fits the available space regardless of the device on which page displays (for example, a desktop computer, laptop computer, tablet, or smartphone).

By implementing a responsive design, the user gets the same full experience as they would on larger screens. On smart phones and tablets, the layout can adjust to the size of the specific device. During this resizing process, elements shift position, re-size, or become hidden. The goal of responsive design is to present all essential content in a user friendly way for all possible screen sizes. Keep in mind, that responsive design is not just a matter of picking the correct set of templates. As the application developer, you are responsible for using the templates and the available components to design a page that is truly responsive. The Universal Theme is an example of a responsive user interface theme.

About the Universal Theme

When you create a new application, the Create Application Wizard uses the Universal Theme.

Key advantages of the Universal Theme include:

- **Responsive Design** - Designed to work just as well on small screen devices (such as smartphones and tablets) as it does on larger screen devices (including laptops and desktops). The UI components in Universal Theme work across varying screen resolutions while maintaining the same or similar functionality. In addition, Universal Theme takes full advantage of ultra high screen resolutions by utilizing vector graphics where possible, and relying upon CSS3 features for UI styling.
- **Versatile User Interface** - Provides all the components and building blocks necessary to build practically any type of business application user interface. To browse all of the components provided with Universal Theme go to the *Universal Theme* application at <https://apex.oracle.com/ut> and select **Components**.
- **Easy Customization** - Effortlessly customize and fully control the look and feel of your applications without becoming an expert in UI design, HTML, CSS, or JavaScript. Using Theme Roller and Template Options, you can easily customize your application to fit your company's brand and customize the look and feel of various components using Template Options.

The *Universal Theme* includes support for theme styles. A theme style is a CSS style sheet that is added to the base CSS. Developers can change the appearance of an application by altering the theme style using the Theme Roller utility.

About Migrating Existing Mobile Applications to the Universal Theme

jQuery Mobile and the jQuery Mobile User Interface used in previous releases have been desupported. If you have an existing mobile application that uses the jQuery Mobile User Interface, you should migrate your existing application to the Universal Theme.

Tip:

To learn more about migrating existing applications to the Universal Theme, go to the *Universal Theme* application at <https://apex.oracle.com/ut> and select **Migration Guide**.

See Also:

- "About Switching the Active Theme"
- "Understanding Template Options"
- "Using Theme Styles and Theme Roller"

Using Theme Styles and Theme Roller

A theme style is a CSS that is added to the base CSS. Using the Theme Roller utility, developers can change the appearance of an application. Universal Theme - 42 includes theme styles.

- [About Theme Styles](#)
- [Creating a Theme Style](#)
- [Editing a Theme Style from the Themes Page](#)
- [Editing a Theme Style from the User Interface Page](#)
- [Enabling Users to Select a Theme Style](#)
- [Using Theme Roller](#)

About Theme Styles

A theme style defines a CSS style sheet that is added to the base CSS to alter the look and feel of an application. Newer themes such as *Universal Theme - 42* can have a base CSS file plus the theme style CSS file. The theme style CSS file are referenced in the page template using the #THEME_STYLE_CSS# substitution string. The Application Express engine replaces this substitution string with the CSS file references defined in the theme style attributes. Use theme styles to customize themes, to switch to a different color scheme, apply a flat look, or make a theme responsive. A theme can have multiple theme styles with one style set as active. You can modify a theme style CSS file using Theme Roller.

Once defined, developers can select a theme style by:

- Editing the Theme Styles attribute when running the Create Application Wizard. See "[Understanding Page Types, Features, and Settings](#)."
- Editing the User Interface. See "[Editing a Theme Style from the User Interface Page](#)."
- Accessing the Create/Edit Theme page. See "[Editing a Theme](#)" and "[Creating a Theme Style](#)."

Creating a Theme Style

To create a theme style:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.
2. Select a theme and click the **Styles** tab.
3. Click **Add Style**.

The Theme Styles page appears.
4. For Settings:

- a. Name - Provide a short descriptive name for the theme style.
- b. Is Current - Select whether this style is the current style used by the theme.
- c. Is Public - Select whether this style can be chosen by end users.
- d. File URLs - Select whether this style is the current style used by the theme.

To learn more about an attribute, see field-level help.

5. For Theme Roller Attributes:
 - a. Read Only - Select **Yes** to prevent any updates to the theme style. Select **No** to enable the theme style to be edited and overwritten using Theme Roller.
 - b. Input Parameter File URLs - Enter the LESS file URLs that will be used to generate this theme style when using Theme Roller.
 - c. Output CSS File URL- Enter the URL for the CSS file that will be generated by Theme Roller for this theme style. .
 - d. Theme Roller JSON Configuration, JSON Configuration - The JSON configuration is generated by Theme Roller when the theme style is saved.

Oracle does not recommend manually updating the configuration. However, you can copy a JSON configuration from another theme style, such as a theme style in another application, to manually update the configuration.
6. Click **Create**.

Editing a Theme Style from the Themes Page

To edit a Theme Style:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.
2. Click the theme name.
3. Select a theme and click the **Styles** tab.
4. Select a theme style.
The Theme Styles page appears.
5. To make changes to the current theme style, edit the appropriate attributes and click **Apply Changes**.
To learn more about an attribute, see field-level help.
6. To delete the current theme style, click **Delete**.
7. Click **Create**.

Editing a Theme Style from the User Interface Page

To edit a Theme Style from the User Interface page:

1. Navigate to the User Interface page:

- a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, click **User Interface Attributes**.
The User Interface page appears.
2. Locate the user interface (for example, **Desktop**) and click the Theme Style name.
The Theme Styles page appears.
 3. To make changes to the current theme style, edit the appropriate attributes and click **Apply Changes**.
To learn more about an attribute, see field-level help.
 4. To delete the current theme style, click **Delete**.
 5. Click **Create**.

Enabling Users to Select a Theme Style

Developers can enable users to select a theme style in a running application. Once enabled, a Customize link appears in the running application, typically in the lower left corner. When the user clicks the Customize link a pop-up displays listing available theme styles. When a user selects a theme style, the theme style is stored persistently as a user preference and will be retained during future sessions.

To enable users to select a theme style:

1. Create an application which has at least two theme styles.
2. Go to the application User Interface page:
 - a. Click **Shared Components**.
 - b. Under User Interface, click **User Interface Attributes**.
The User Interface Attributes page appears. Defined User Interfaces display at the top of the page.
3. Edit the User Interface Details.
 - a. Click the **Edit** icon adjacent to the appropriate user Interface (for example, Desktop).
 - b. Under Attributes, set **Enable End Users to choose Theme Style** to **Yes**.
 - c. Click **Apply Changes**.
4. Edit each theme style and set the **Is Public** attribute to **Yes**.
 - a. On the User Interface page, click the theme style name (for example, Vita).
 - b. Under Settings, set **Is Public** to **Yes**.
 - c. Click **Apply Changes**.
The User Interface page reappears.

Using Theme Roller

Theme Roller is a live CSS editor that enables developers to quickly change the colors, rounded corners and other attributes of their applications without touching a

line of code. Theme Roller displays in the Runtime Developer Toolbar if at least one of your theme styles has the **Input Parameter File URLs** attribute defined.

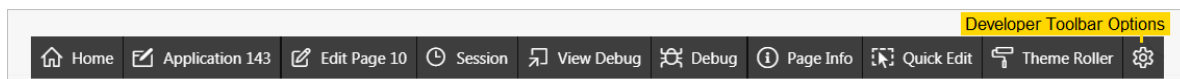
 **Tip:**

To see an example of Theme Roller, install and run a productivity or sample application that uses the Universal Theme - 42."Utilizing the App Gallery."

To use Theme Roller:

1. Preview the page by running it.

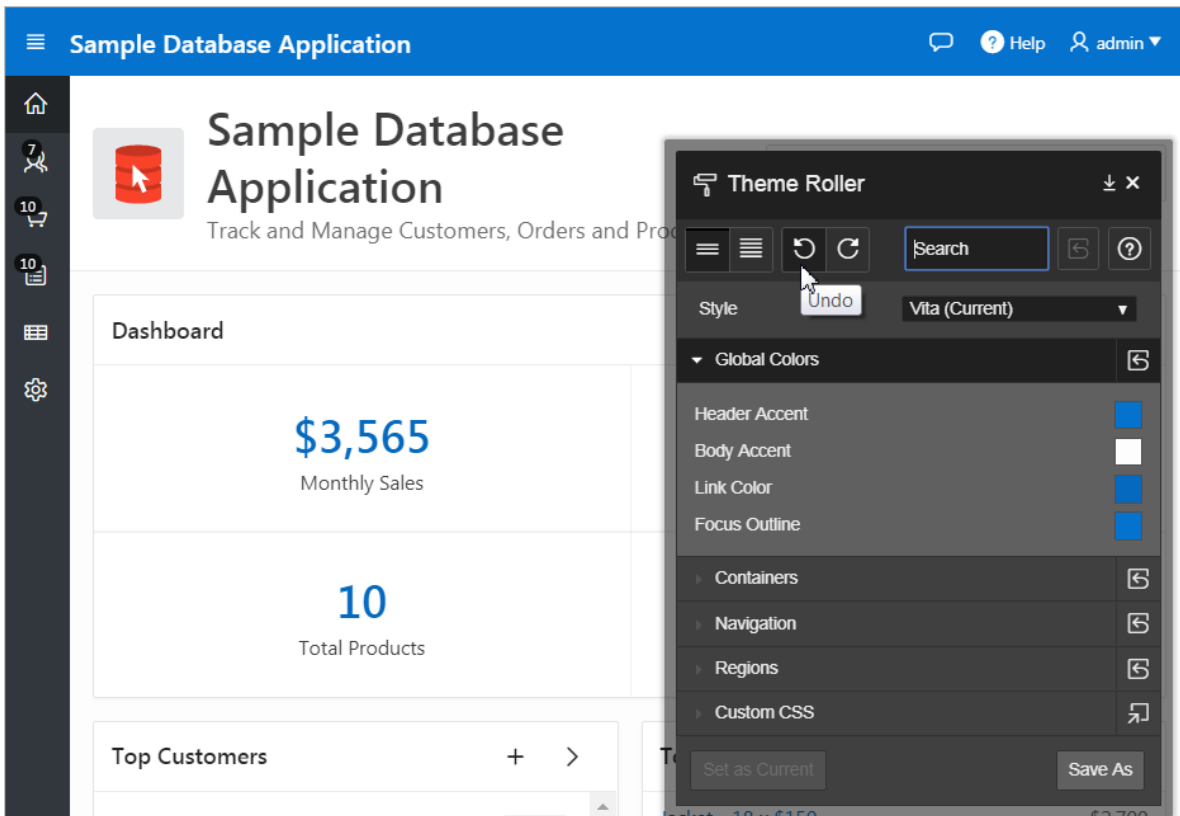
When a developer run a desktop application, the Runtime Developer toolbar displays at the bottom of any editable running page.

 **Tip:**

Theme Roller only displays in the Runtime Developer Toolbar if theme styles have been defined. See "[Editing a Theme](#)" and "[Styles](#)."

2. Click **Theme Roller** on the Runtime Developer Toolbar.

Theme Roller fetches the styles for your application and loads them in the editor.



Hover over active icons and menus to view their names.

3. You can expand or collapse sections by clicking on the section names. Primary control include:
 - **Show Common** - Displays all attributes.

 **Tip:**

Altering the Custom CSS section can override the theme style of Universal Theme. Be careful with the selector you use in this section. Hiding or obscuring content may degrade your user's experience.

- **Show All** - Displays all attributes.
- **Undo** - Reverts the currently edited theme to the previous action. This feature will not work if you have switched themes.
- **Redo** - Dismisses the current Undo revision and goes to the next one in the history. This feature will not work if you have switched themes.
- **Search** - Search for properties, groups, and colors. As you search, Theme Roller automatically displays only properties that match your search string.
- **Reset** - Reverts the selected theme to the last version saved on the server. After resetting your theme and reloading your page, you can use Undo and Redo to restore your changes.
- **Help** - Displays a Help window.

4. From Style, select a new style, choose an existing style from the list.
5. Under Global Colors, click the color swatch to select new colors.
6. To edit a specific component, expand a group and select new colors or styling of the component you wish to edit.

For attributes that support the selection of foreground and background colors, the check box preceding the two color selections indicates color contrast information based on Web Content Accessibility Guidelines (WCAG) 2.0. A higher calculated contrast score indicates that the color combination is more accessible.

**Tip:**

to set the theme style you are working as the current one for the application, click **Set as Current**.

7. To set the theme style you are working as the current one for the application.
8. Click **Save** to commit your changes to the server. If the selected theme is read only, then click **Save As** to save your changes as a new theme.

Understanding Template Options

Template options provide developers with a declarative approach to applying different styles to components on an Oracle Application Express page.

- [About Template Options](#)
- [About Default Template Options](#)
- [About Template Option Groups and Presets](#)

**See Also:**

["Using Template Options"](#) and ["Using Quick Edit to Modify Live Template Options"](#)

About Template Options

Template options enable developers to declaratively apply CSS modifiers to the templates they have chosen for pages, regions, reports, lists, breadcrumbs, items and buttons. A CSS modifier is a reference to a CSS class defined in a CSS style sheet. CSS modifiers enable a developer to use the same HTML markup but present it in a variety of different styles simply by applying a CSS. Template options reduce the need to have a large number of nearly identical templates defined in a theme in order to achieve different styles for button colors, region widths, font settings, item label markup, and so on.

Examples of how developers can use template options include:

- Applying different colors or accents
- Applying different spacing and padding

- Rendering buttons in different styles, with and without icons
- Displaying form fields with different alignments.

Without template options, applying these variations would require a large number of nearly identical templates, or in-depth knowledge of CSS. Components that support template options include: pages; regions; classic reports; breadcrumbs; lists; items and labels; and buttons.

About Default Template Options

Developers can select template options that are not part of a template option group to be the default for a template. App Builder automatically applies default template options by rendering the component that references the template. Default template options are not written to an actual component. Instead, a `#DEFAULT#` substitution string is defined for the component, thus enabling developers to centrally modify the template option defaults.

Developers can choose not to use the template option defaults, which removes the `#DEFAULT#` substitution string from the component. If template option defaults are disabled, then developers can selectively apply available template options to a component. All template options that are selected directly when editing the component are written to the component's metadata, represented by their corresponding CSS classes string. At runtime, the selected template options and default template option are part of the default are combined with the component's CSS classes string and applied to the component.

About Template Option Groups and Presets

Developers can define the purpose of related template options by creating template option groups. Examples of template option groups include Button Size, Button Style, List Examples, List Style, and Form Label Position. When developers create a template option group, they can specify a template option preset at the template-level. For example, suppose you have a button template that has the template options *Large* and *Small*. If you define *Small* as the preset, then any new button referencing the template would automatically have the template option *Small* applied to it when is it created. Template options that are part of a group are optional unless a present is defined at the template-level.

Accessing the Themes Page

Manage themes on the Themes page.

Tip:

You can change the selected default templates for a theme on the Create/Edit Theme page. See "[Editing Themes](#)".

To access the Themes page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

3. Click **Shared Components**.
4. Under User Interface, select **Themes**.
The Themes page appears.
5. Click the **View Report** icon.
The Themes page appears as a report. A check mark in the Is Current column indicates which theme is selected.
6. Click the theme name.
The Edit Theme page appears.

Editing Themes

Edit a theme by editing theme attributes.

- [Editing a Theme](#)
- [Theme Attributes](#)



See Also:

["Creating Custom Themes"](#)

Editing a Theme

To review or edit a theme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Themes**.
The Themes page appears.
5. On the Themes page, click the **View Report** icon.
The Themes page appears as a report. A check mark in the Is Current column indicates which theme is selected.
6. Click the theme name.
The Create/Edit Theme page appears and is divided into sections.
7. Edit the appropriate Theme attributes as described in "[Theme Attributes](#)."
8. To save any changes, click **Apply Changes**.

Theme Attributes

Learn about Theme attributes available on the Create/Edit Theme Attributes page. To learn more about an attribute and view examples, see field-level Help.

- [Name](#)

- [Theme Subscription](#)
- [JavaScript and Cascading Style Sheets](#)
- [Component Defaults](#)
- [Region Defaults](#)
- [Dialog Defaults](#)
- [Transition Defaults](#)
- [Global Templates Options](#)
- [Icons](#)
- [Image](#)
- [Styles](#)
- [Files](#)

Name

[Table 12-1](#) describes the attributes under Name.

Table 12-1 Name

| Attribute | Description |
|-------------------------------|--|
| Application | Identifies the current application ID. See Also: " Editing Application Attributes " |
| Theme Number | Identifies the number of a theme. A theme is a collection of templates which define the user interface of an application. |
| Name | Enter a short descriptive name for the theme. |
| User Interface | Specifies the user interface for which the theme is designed. See Also: " User Interface Details Page " |
| Navigation Type | Identifies the selected Navigation Type. Valid options include: Tabs or List. See Also: " Creating Tabs " and " Creating Lists " |
| Navigation Bar Implementation | Identifies the selected Navigation Bar Implementation. Valid options include: Classic or List. See Also: " User Interface Details Page " |
| Description | Enter a description for the theme. |

Theme Subscription

Displays themes to which this application subscribes. When an application subscribes to a theme, all theme attributes, subscribed template options, and subscribed templates are set to read-only.



See Also:

["Understanding Theme Subscriptions"](#)

JavaScript and Cascading Style Sheets

[Table 12-2](#) describes the attributes under JavaScript and Cascading Style Sheets.

Table 12-2 JavaScript and Cascading Style Sheets

| Attribute | Description |
|----------------------|---|
| JavaScript File URLs | Enter JavaScript file URLs for code to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file, you can use the substitution string #MIN# to include .min or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. JavaScript file URLs you enter here replaces the #THEME_JAVASCRIPT# substitution string in the page template To view examples, see field-level Help. |
| CSS File URLs | Enter Cascading Style Sheet file URLs to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string #MIN# to include .min or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string #APP_VERSION# if you want to include the application's version in the file URL. File URLs you enter here will replace the #THEME_CSS# substitution string in the page template. To view examples, see field-level Help. |

Component Defaults

[Table 12-3](#) describes the default templates by component type.

Table 12-3 Component Default Templates

| Attribute | Description |
|-------------------------------|--|
| Page | Identifies the default template for displaying pages. If a developer does not explicitly choose a template, then the Application Express engine uses the template specified here. |
| Navigation Bar List | Specifies the default navigation bar list template used when you define the navigation bar type as List in your application. If a developer does not explicitly choose a template, then the Application Express engine uses the template specified here. |
| Navigation Menu List Position | Specifies the default navigation menu list position. |
| Navigation Menu List (Top) | Specifies the default navigation menu list template used when you create a new list to be displayed on the top of the page. |
| Navigation Menu List (Side) | Specifies the default navigation menu list template used when you create a new list to be displayed on the side of the page. |
| Login Page | Specifies the default template for displaying pages. If a developer does not explicitly choose a template then the Application Express engine uses the template specified here. |

Table 12-3 (Cont.) Component Default Templates

| Attribute | Description |
|-----------------------|--|
| Error Page | Optional. Specifies a page template to use for errors that display on a separate page as opposed to those that display inline. Leave this attribute blank if you do not wish to use a template designed to display errors. This setting only applies to Normal pages. |
| Printer Friendly Page | Identifies the template to be used when the Application Express engine is in printer friendly mode. When calling the Application Express to render a page, you have the option to identify a printer friendly attribute with values of Yes or No . If you select Yes , then the page displays using a printer friendly template. The Application Express engine displays all text within HTML form fields as text. The printer friendly template does not need to have the #FORM_OPEN# or #FORM_CLOSE# tags. The objective is to be able to display information with few tables and in a format suitable for printing. |
| Breadcrumb | Identifies the default breadcrumb template used when you create a breadcrumb. |
| Button | Identifies the default button template used when you create a button. |
| Legacy Calendar | Specifies the default calendar template used when you create a calendar. |
| Default Label | Specifies the default label template used when you create new label |
| Optional Label | Specifies the default optional label template used when you create new label. |
| Page Template Options | Template options allow for selecting a number of CSS customization settings to be applied to the current page template. Template options and their corresponding CSS classes are defined at the page template level. The page template needs to have the substitution string #PAGE_CSS_CLASSES# defined in either the page header or body. The string #PAGE_CSS_CLASSES# is substituted for any manually entered page CSS classes, which are combined with the CSS classes defined by the selected template options. |
| Required Label | Specifies the default required label template used when you create new label. |
| List | Specifies the default list template used when you create a list. |
| Region | Specifies the default region template used when you create a region. |
| Classic Report | Identifies the default region template used when you create a report. |
| Header Toolbar | Identifies the default header toolbar region template used when you create a new page that includes a header or footer tool bars. |
| Footer Toolbar | Identifies the default header toolbar region template used when you create a new page that includes a header or footer tool bars. |

Region Defaults

[Table 12-4](#) describes the default templates available under Region Defaults.

Table 12-4 Region Defaults

| Attribute | Description |
|---------------------|--|
| Breadcrumb | Identifies the default breadcrumb template used when you create a breadcrumb. |
| Charts | Specifies the default chart template used when you create a chart. |
| Forms | Specifies the default form template used when you create a form. |
| Lists | Specifies the default region template used when you create a list. |
| Reports | Specifies the default region template used when you create a report. |
| Tabular Forms | Specifies the default region template used when you create a tabular form. |
| Wizards | Specifies the default region template used when you create a new wizard component. |
| Interactive Reports | Specifies the default region template used when you create an interactive report |

Dialog Defaults

[Table 12-5](#) describes the default templates available under Dialog Defaults.

Table 12-5 Dialog Defaults

| Attribute | Description |
|-----------------------|---|
| Dialog Content Region | Specifies the default region template used when you create a content region on a Dialog using the Create Page Wizard and Create Application Wizard. |
| Dialog Button Region | Specifies the default region template used when you create a button region on a Dialog using the Create Page Wizard and Create Application Wizard. |
| Dialog Page | Specifies the default template for displaying modal or non-modal dialog pages. If a developer does not explicitly choose a template then the Application Express engine uses the template specified here. This template will also be used for errors that display on a separate page as opposed to those that display inline. |

Transition Defaults

Transition Defaults is only visible for a Mobile theme. [Table 12-6](#) describes attributes available under Transition Defaults.

Table 12-6 Transition Defaults

| Attribute | Description |
|------------------|--|
| Page Transition | Specifies the transition which should be used when navigating to a new page or when the page is getting submitted. |
| Popup Transition | Specifies the transition which should be used when a popup is opened. |

Global Templates Options

Global template options are defined at the theme-level and are available for all components of a given type.



See Also:

"[Understanding Template Options](#)" and "[Managing Global Template Options](#)"

Icons

[Table 12-7](#) describes attributes under Icons.

Table 12-7 Icons

| Attribute | Description |
|-----------------------------|---|
| Library | Select the icon library that is loaded when Oracle Application Express displays a page. The icons within the library are also be listed in the picker for Region and Button Icon CSS Classes, implemented within Page Designer. |
| Custom Library File URLs | Enter Cascading Style Sheet file URLs for custom icons to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string <code>#MIN#</code> to include <code>.min</code> or <code>#MIN_DIRECTORY#</code> to include <code>minified/</code> in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string <code>#APP_VERSION#</code> if you want to include the application's version in the file URL. File URLs you enter here will replace the <code>#THEME_CSS#</code> substitution string in the page template. To view examples, see field-level Help. |
| Custom Classes | Specify a comma delimited list of CSS class names which is listed in the picker for Region and Button Icon CSS Classes, implemented within Page Designer. |
| Custom Prefix Class | Specify a CSS class that will at runtime prefix the Icon CSS Classes defined for Regions, Buttons and Lists, if they are not using a class of the Icon Library. |
| Date Picker Icon Name | Enter the name and location of the image to be used when items of type Date Picker are displayed. For example: <code>#IMAGE_PREFIX#new_cal.bmp</code> |
| Date Picker Icon Attributes | Enter the image attributes for the Calendar Icon. For example: <code>width="16" height="16" alt="Display Calendar"</code> |

Image

Use the **Custom Image** attribute to specify whether this theme uses a default theme image or an uploaded theme image. Select **Yes** or **No**.

Styles

Displays theme styles available to the current theme. Theme Styles define the CSS for your application's theme and control the user interface of your application. Theme styles are referenced in the page template using the `#THEME_STYLE_CSS#` substitution string. The Theme Style currently used displays a check mark under **Is Current**.

 **See Also:**

["Using Theme Styles and Theme Roller"](#)

Files

The **File Prefix** attribute displays the virtual path the Web server uses to point to the files of the theme. Do not specify anything to reference files which are stored with your theme definition in the database. For performance reasons you can also store your theme files on your Web Server. Use `#IMAGE_PREFIX#` or any valid URL to reference them under Styles.

 **See Also:**

["Managing Static Application Files"](#) and ["Managing Static Workspace Files"](#)

Managing Themes and Subscriptions

Developers can manage themes directly or through the use of theme subscriptions.

- [Understanding Theme Subscriptions](#)
- [Creating a Theme](#)
- [Switching Themes](#)
- [Changing a Theme Identification Number](#)
- [Copying a Theme](#)
- [Deleting a Theme](#)
- [Creating a Master Theme Application](#)
- [Managing Workspace Themes](#)
- [Managing Instance Themes](#)
- [About Exporting and Importing Themes](#)
- [Viewing Theme Reports](#)

Understanding Theme Subscriptions

Developers can manage themes using of theme subscriptions.

- [About Theme Subscriptions](#)
- [About Modifying a Theme Subscription](#)
- [Verifying if a Theme Refresh Is Needed](#)
- [Refreshing a Subscribed Theme](#)
- [Unsubscribing to a Theme](#)
- [Re-subscribing to a Built-in Theme](#)

About Theme Subscriptions

Subscribing to a master theme enables the theme to be upgraded during future Oracle Application Express releases.

Developers subscribe to a theme when:

- **Running the Create Application Wizard.**
When a developer runs the Create Application Wizard and selects a built-in theme, the theme is automatically subscribed to a master theme. As an alternative to using a built-in theme, the Create Application Wizard provides the option to copy a theme from an existing application. When copying a theme from an existing applications, developers can choose if they wish to have this theme be subscribed to. See "[Understanding Page Types, Features, and Settings](#)."
- **Creating a new theme from the Theme repository.**
When you create a new theme the associated application automatically subscribes to it. See "[Creating a Theme](#)."

Once an application subscribes to a master theme only the default templates can be changed. All theme attributes, subscribed template options, and subscribed templates are set to read-only. In other words, only the default templates can be changed. The advantage of subscribing to a master theme is that the theme can be upgraded during future Oracle Application Express releases. When a theme is subscribed to, developers cannot modify the theme or template metadata when they access it from the theme or template edit pages.

About Modifying a Theme Subscription

To modify a template without breaking a subscription to the master theme, a developer can create a local copy of the template in their workspace. Local template copies are not subscribed to and are therefore editable. You can take the same approach with theme styles. You can create a local copy of a theme style and then add it to a subscribed theme. Theme styles created locally are editable but will be excluded from theme refreshes.



See Also:

"Copying a Theme"

Verifying if a Theme Refresh Is Needed

The Verify Theme Subscription Wizard reviews all theme attributes, templates, and template options in the subscribing theme and raises errors for those that are not in sync with the master theme. The wizard also highlights any templates that are defined in the master theme but not yet published to the subscribing theme.

To run the Verify Theme Subscription Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Themes**.

The Themes page appears.

5. On the Themes page, click the **View Report** icon.

The Themes page appears as a report. A check mark in the Is Current column indicates which theme is selected.

6. Select the theme.

The Create/Edit Theme page appears.

7. Locate Theme Subscription and click **Verify**.

8. Follow the on-screen instructions.

To publish changes from the master theme to the subscribing theme, click **Refresh** at the end of the wizard.

Refreshing a Subscribed Theme

To refresh a subscribed theme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Themes**.

The Themes page appears.

5. On the Themes page, click the **View Report** icon.

The Themes page appears as a report. A check mark in the Is Current column indicates which theme is selected.

6. Select the theme.

The Create/Edit Theme page appears.

7. Locate Theme Subscription and click **Refresh Theme**.

 **Tip:**

If the theme is not subscribed, the Refresh Theme button does not appear. See "[Re-subscribing to a Built-in Theme.](#)"

8. Click **Apply Changes**.

Unsubscribing to a Theme

 **Note:**

Unsubscribing to a theme prevents future theme updates. Be aware that re-subscribing to a theme is multiple step process that involves copying and switching themes. To learn more, see "[Re-subscribing to a Built-in Theme.](#)"

Unsubscribing a theme also removes the subscription from all templates, styles, files and so on that are part of the theme.

To unsubscribe to a theme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Themes**.
The Themes page appears.
5. On the Themes page, click the **View Report** icon.
The Themes page appears as a report. A check mark in the Is Current column indicates which theme is selected.
6. Select the theme.
The Create/Edit Theme page appears.
7. Locate Theme Subscription and click **Unsubscribe**.
A warning dialog displays.
8. Click **Unsubscribe Theme**.

Re-subscribing to a Built-in Theme

If a theme is currently unsubscribed, you can return to a subscribed version of this theme.

To re-subscribe to a theme.

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. From the Tasks list, select **Change Identification Number** and change the theme to a new identification number. See "[Changing a Theme Identification Number](#)."
3. Run the Create Theme Wizard and create the theme again. See "[Creating a Theme](#)."
4. Switch the current application to the use the new theme. See "[Switching Themes](#)."
5. Delete the original theme from which you created a new identification number. See "[Deleting a Theme](#)."

Creating a Theme

Only themes currently associated with an application display on the Themes page. To access other themes, you must run the Create Theme Wizard. Once you create a theme, you can access it on the Themes page. You can create a theme from scratch or select an existing theme from the repository.

Tip:

When you create a new theme, the associated application automatically subscribes to it. See "[Managing Themes and Subscriptions](#)."

To create a theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. Click **Create**.
3. Select a creation method:
 - **From the Repository** - View and select a theme from the repository.
 - **As a copy from another application** - Select the application and then the theme.
 - **From Scratch** - Create a theme from scratch.
 - **From Export** - Create a theme to import from the export repository.
4. Follow the on-screen instructions.

To learn more about an attribute, see field-level Help.



See Also:

["Editing Themes"](#)

Switching Themes

This section describes how to change the active theme for an application.

- [About Switching the Active Theme](#)
- [Switching an Active Theme](#)

About Switching the Active Theme

When you switch to a theme, all components with assigned templates are assigned to a corresponding template in the theme. App Builder accomplishes template mapping through the assignment of template class identifiers.

When switching an active theme, remember:

- You can only switch to a theme if the theme exists. For example, before you can switch to a theme available in the repository, you must first create it. See "[Creating a Theme](#)".
- If the wizard locates multiple matches (that is, when several templates share the same template type and templates class), the wizard defaults to a template matching type, class, and name. If no match is found, you must select another template with the same type and class.
- If no matching template is found (that is, no template with the same type and class exists) then you are prompted to select a template with the same type but with a different class. Then, a warning displays explaining this issue might lead to problems with page functionality, or issues with the page display. When no template of the same template type exists, then the theme cannot be switched. You are blocked from switching themes.

Switching an Active Theme




Tip:

To complete this procedure, you must have at least two themes installed. If you do not have at least two themes installed, on the Themes page click **Create Theme**. You can create a theme from scratch or simply select an existing theme from the repository. See "[Creating a Theme](#)."

To switch an active theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.
 - The Themes page appears.
 2. Click **Switch Theme**.
 - The Switch Theme page appears.
 3. For Identify Theme:
 - a. Currently Active Theme - Select the current theme for the application. A theme identifies a collection of templates which define the look and feel of the application.
 - b. Switch to Theme - Select the new theme for the application.
 - c. Click **Next**.
 - The Verify Compatibility page appears.
 4. Review the Status column to identify problematic mappings:
 - A **check** indicates the mapping was successful.
 - A **warning** indicates there are 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 App Builder was unable to map the class among the themes. Ensure that a class is identified for the templates in both themes.
-  **Note:**

Failure to resolve warnings may result in a loss of functionality and applications that do not display well.
5. Click **Next** to continue.
 6. Click **Switch Theme**.

Changing a Theme Identification Number

Each theme has an identification number (ID). You can use the Change Theme ID utility to change a theme ID to another identification number. Changing a theme ID is useful when you want to export a theme with a different number and then import it into another application.

To change a theme identification number:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.
- The Themes page appears.

2. On the Tasks list, click **Change Identification Number**.
3. For Theme Number:
 - a. Identify Theme - Select a theme.
 - b. New Theme Identification Number - Specify an identification number.
 - c. Click **Next**.
4. Confirm your changes and click **Change Theme ID**.

Copying a Theme

Each theme is identified by a numeric identification number (ID). When you copy a theme, you specify a theme ID. Copying a theme is useful when experimenting with editing a theme or to export a theme with a different ID.

To copy a theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. On the Tasks list, click **Copy Theme**.
3. For Theme Number:
 - a. Application - Identifies the current application.
 - b. Copy From Theme - Select the theme you want to copy.
 - c. Copy to this Theme ID - Enter a new ID for the theme.
 - d. Subscribe Theme - Select **Yes** or **No**.
 - e. Click **Next**.
4. Confirm your changes and click **Copy Theme**.



See Also:

["Understanding Theme Subscriptions"](#)

Deleting a Theme

You can only delete inactive themes. When you delete a theme, App Builder only removes inactive templates.

 **Tip:**

Themes are tied to an application. If you modify a theme, those changes only apply to the application being edited. If you delete a theme, any template modifications are lost. To keep template modifications, you must export the theme. See "[Exporting Themes](#)."

To delete a theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. On the Tasks list, click **Delete Theme**.
3. From Delete Theme, select the theme you want to delete and click **Next**.
4. Click **Delete Theme**.

Creating a Master Theme Application

Developers can create a master theme application. As a best practice, Oracle recommends starting with an existing productivity or sample application that uses *Universal Theme - 42* (for example, *Sample Database Application*).

- [Creating a Master Theme Application by Creating a New Application](#)
- [Creating a Master Theme Application from an Existing Application](#)

Creating a Master Theme Application by Creating a New Application

To create a master theme application by creating a new application:

1. Create a new application by running the Create Application Wizard:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Click the **Create** button.
2. Select **Copy an existing application** and select an application that uses *Universal Theme - 42* (for example, *Sample Database Application*).
3. Follow the on-screen instructions.

Creating a Master Theme Application from an Existing Application

To create a master theme application from an existing application:

1. Copy the theme from an application that uses *Universal Theme - 42* (for example, *Sample Database Application*) and subscribe to it. See "[Copying a Theme](#)."
2. Switch to the new theme. See "[Switching Themes](#)."

3. Follow the on-screen instructions.

Managing Workspace Themes

Workspace administrators manage the theme repository for a workspace. Workspace administrators can add a theme to the repository, making it available to all developers within a workspace, or delete it.

- [Adding a Workspace Theme to the Theme Repository](#)
- [Viewing an Existing Workspace Theme](#)
- [Deleting a Workspace Theme](#)
- [Modifying a Workspace Theme](#)
- [Exporting a Workspace Theme](#)



See Also:

"Managing Public Themes" in *Oracle Application Express Administration Guide*

Adding a Workspace Theme to the Theme Repository

To add a theme to the Theme repository:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. On the Tasks list, click **Manage Workspace Themes**.
3. Select **Create Workspace Theme** and click **Next**.



Tip:

If no workspace themes exist, the wizard selects **Create** for you.

4. For Create Workspace Theme:
 - a. Application - Select the application.
 - b. Application Theme to Copy - Select the theme to copy to the workspace theme repository.
 - c. Subscribe Theme - Select whether to subscribe the workspace theme to the selected application theme. This option enables you to refresh the workspace theme with changes made to the selected application theme.

- d. Click **Next**.
 5.
 - a. Theme Number - Enter a number with a value greater than 100. This number uniquely identifies this theme within a workspace.
 - b. Theme Name - Enter a theme name. This name can be the same or different from the application theme name.
 - c. Description - Enter a description of the theme.
 - d. Click **Next**.
 6. Click **Create Workspace Theme**.

Viewing an Existing Workspace Theme

To view an existing workspace theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. From the Tasks List, click **Manage Workspace Themes**.

The Workspace Themes page appears.

Tip:

You cannot edit a workspace theme directly. To modify a workspace theme, you must create an application using the theme, modify it, and then manually add it to the workspace theme repository. See "[Modifying a Workspace Theme](#)."

Deleting a Workspace Theme

To delete a workspace theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. From the Tasks List, click **Manage Workspace Themes**.

The Workspace Themes page appears.

3. Select the theme name.

4. Click **Delete**.
5. Confirm your delete request and click **OK**.

Modifying a Workspace Theme

You cannot edit a workspace theme directly. To modify a workspace theme, you must create an application using the theme, modify it, and then manually add it to the workspace theme repository.

To modify a workspace theme:

1. Create an application using the theme you want to modify. See "[Creating Database Applications](#)."
2. Modify the theme. See "[Editing Themes](#)."
3. Delete the existing workspace theme. See "[Deleting a Workspace Theme](#)."
4. Add the modified theme to the workspace theme repository. See "[Adding a Workspace Theme to the Theme Repository](#)."

Exporting a Workspace Theme

You export a theme in the same way you export any related application files. Exporting a workspace theme involves the following steps:

1. Create an application using the theme. See "[Creating Database Applications](#)."
2. Export the application. See "[Exporting an Application](#)."
3. Import the exported file into the target Oracle Application Express instance. See "[Importing an Application, Page or Component Export](#)."
4. Install the exported file from the Export Repository. See "[Installing Export Files](#)."

Managing Instance Themes

Instance administrators manage the theme repository for an entire Oracle Application Express instance. Only an Instance administrators can add or delete a theme from the repository.



See Also:

"Managing Public Themes" in *Oracle Application Express Administration Guide*

About Exporting and Importing Themes

You export a theme in the same way you export any related application files. Exporting a theme from one development instance to another involves the following steps:

1. Create an application using the theme. See "[Creating Database Applications](#)."
2. Export the theme. See "[Exporting Themes](#)."

3. Import the exported file into the target Oracle Application Express instance. See "[Importing Export Files](#)."
4. Install the exported file from the Export Repository. See "[Installing Export Files](#)."

Viewing Theme Reports

Access the following reports to better manage themes and templates.

- [Viewing All Templates in a Theme](#)
- [Viewing Theme Template Counts](#)
- [Viewing File References](#)
- [Viewing Class References](#)
- [Viewing Template Substitution Strings](#)

Viewing All Templates in a Theme

To view all templates that comprise a theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. Click **Reports**.
3. On the Theme Reports page:
 - a. Report - Select **Application Templates**.
 - b. Theme - Select either **42. Universal Theme** or **All Themes**.
 - c. Click **Go**.

A list of templates appears with the template type, template name, theme, and template class.

4. To edit a template, select the template name.

Viewing Theme Template Counts

The Theme Template Count report lists which template classes currently have templates created for them.

To view the Theme Template Count report:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. Click **Reports**.
3. On the Theme Reports page:
 - a. Report - Select **Theme Template Counts**.
 - b. Theme - Select a theme or select **All**.
 - c. Click **Go**.

Viewing File References

The File References report displays a list of all files associated with templates, shared components, or page components in the current application.

To view the File References report:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. Click **Reports**.
3. On the Theme Reports page:
 - a. Report - Select **File References**.
 - b. Theme - Select a theme or select **All**.
 - c. Click **Go**.
4. On the File References page:
 - a. Show - Select the type of component to include in the report. If you do not make a selection, no results are returned.
 - b. Show Files - Select one of the following:
 - **With context** displays the component, the theme identification number, the component name, the image (if applicable), and the page number. Select the page number.
 - **Without context** displays only the file name and the image (if applicable).
 - c. File Extensions - Select the type of extensions for which to search.
 - d. Click **Go**.
5. To download a comma-delimited file (.csv) version of this report, click **Download** at the bottom of the page.

Viewing Class References

Accessing the Class References report displays a list of classes associated with templates, shared components, or page components in the current application.

To view the Class References report:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. Click **Reports**.
3. On the Theme Reports page:
 - a. Report - Select **Class References**.
 - b. Theme - Select a theme or select **All**.
 - c. Click **Go**.
4. On the Class References page:
 - a. Show - Select the components to check for a class reference. If you do not make a selection, no results are returned.
 - b. Show Class Names - Select one of the following:
 - **With context** displays the component, the theme identification number, the component name, the image (if applicable), and the page number.
 - **Without context** displays only the referenced class.
 - c. Click **Go**.
5. To download a comma-delimited file (.csv) version of this report, click **Download** at the bottom of the page.

Viewing Template Substitution Strings

Use the Template Substitution Strings report to view all supported substitution strings by component.

To view the Substitution String report:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. Click **Reports**.
3. On the Theme Reports page:
 - a. Report - Select **Template Substitution Strings**.
 - b. Theme - Select which themes to include in the report.
 - c. Click **Go**.
4. To link to a template definition, select the component name.



See Also:

["Using Substitution Strings"](#)

Creating Custom Themes

Create a custom theme by modifying an existing templates.

The Application Express engine creates an application user interface based on a named collection of templates called a **theme**. Templates control the look and feel of the components in an application. If you must create a custom template, it is generally easier to start with an existing template and then modify it. Once you have created one or more default templates, you can modify those templates to fit your specific needs.

- [About Cascading Style Sheets](#)
- [About Calling the JavaScript File from the Page Template](#)
- [About Using Escaping Syntax in Substitution Strings](#)
- [Selecting a Default Page Template](#)
- [Creating a New Template](#)
- [Using Template Options](#)
- [Viewing Template Reports](#)
- [Managing Templates](#)
- [Breadcrumb Templates](#)
- [Button Templates](#)
- [Legacy Calendar Templates](#)
- [Label Templates](#)
- [List Templates](#)
- [Page Templates](#)
- [Popup LOV Templates](#)
- [Region Templates](#)
- [Report Templates](#)



See Also:

["Using Themes"](#) and ["Managing Themes and Subscriptions"](#)

About Cascading Style Sheets

A Cascading Style Sheet (CSS) provides a way to control the style of a web page without changing its structure. When used properly, 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 display.

When using built-in themes, you can find the theme specific CSS files in the following locations:

```
/i/themes/theme_xx  
/i/themes/theme_xx/css
```

Theme specific image can be found in the following locations:

```
/i/themes/theme_xx  
/i/themes/theme_xx/images
```

Where *xx* is the theme number. Theme specific CSS files include the Oracle Application Express version number to preserve backward compatibility for imported applications using older versions of a theme.

**See Also:**

["Using Custom Cascading Style Sheets"](#)

About Calling the JavaScript File from the Page Template

In Oracle Application Express, you can reference a `.js` file in the page template. This approach makes all the JavaScript in that file accessible to the application. This is the most efficient approach since a `.js` file loads on the first page view of your application and is then cached by the browser.

**See Also:**

["JavaScript"](#)

About Using Escaping Syntax in Substitution Strings

Developers can append an exclamation mark (!) followed by a predefined filter name to substitution strings to escape special characters in the substitution value. Output escaping is an important security technique to avoid Cross Site Scripting (XSS) attacks in the browser.

**See Also:**

["Controlling Output Escaping in Substitution Strings"](#)

Selecting a Default Page Template

This section describes how to select a default page template. You can specify a default page template in two ways:

- Select a default page template within a specific theme.
- Select a specific page template on a page-by-page basis.

By default, the Application Express engine uses the Page template specified on the Themes page.

- [Selecting a Page-level Template Within a Theme](#)
- [Selecting a Page-level Template for a Specific Page](#)

Selecting a Page-level Template Within a Theme

To specify a default page template within a theme:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. Select a theme name.

The Create/Edit Theme page appears.
3. Scroll down to Component Defaults.
4. From Page, make a new selection.
5. Click **Apply Changes** at the top of the page.



See Also:

["Editing Themes"](#)

Selecting a Page-level Template for a Specific Page

To specify a page-level template for a specific page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. To edit the page-level template:
 - a. View the page in Page Designer.
 - b. In the Rendering tab, click the page title.

- c. In the Property Editor, locate the **Appearance**. From Page Template, select a new template.
- d. Click **Save**.

**See Also:**

["Viewing a Page in Page Designer"](#)

Creating a New Template

If you must create a custom template, start with an existing template, copy it, and then modify it. Once you have created one or more default templates, you can modify those templates to fit your specific needs.

To create a custom template:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application to which you want to copy the template.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.
5. Click **Create**.
6. Select the type of template you want to create and click **Next**.
7. Select a creation method:
 - **From Scratch**
 - **As a Copy of an Existing Template**
8. Follow the on-screen instructions.

**Tip:**

Make sure you associate your template with the correct theme.

Using Template Options

Template options provide a declarative way for developers to apply different styles to components on an Oracle Application Express page.

- [Managing Global Template Option Groups](#)
- [Managing Global Template Options](#)
- [Managing Template Options](#)



See Also:

["Understanding Template Options"](#)

Managing Global Template Option Groups

Global template option groups are defined at the theme-level and are available for all components of a given type. This section describes how to create and edit global template option groups.

- [Creating a Global Template Option Group](#)
- [Editing Global Template Option Groups](#)



See Also:

["About Template Option Groups and Presets"](#) and ["Understanding Template Options"](#)

Creating a Global Template Option Group

To create a global template option group:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.

The Themes page appears.

2. Select a theme.
3. Locate **Global Template Options**.
4. Click **Edit Template Option Groups**.

Template Options Groups dialog appears.

5. To create a group:
 - a. Click **Create**.
 - b. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help.
 - c. Click **Create**.

Editing Global Template Option Groups

To edit global template option groups:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. Select a theme.
3. Locate **Global Template Options**.
4. Click **Edit Template Option Groups**.
Template Options Groups dialog appears.
5. To edit a group:
 - a. Click the group name.
 - b. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help.
 - c. Click **Apply Changes**.
6. To delete a group:
 - a. Click the group name.
 - b. Click **Delete**.

Managing Global Template Options

This section describes how to create and edit global template options.

- [Creating Global Template Options](#)
- [Editing Global Template Options](#)



See Also:

["About Template Options"](#)

Creating Global Template Options

To create a global template option:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.

2. Select a theme.
3. Locate **Global Template Options**.
4. Click **Add Template Option**.
Template Options Groups dialog appears.
5. On Template Options:
 - a. Display Sequence - Specify the sequence for this option.
 - b. Template Type - Select the type of template for which the template option is defined.
 - c. Group - Enter a unique group name. Group names enable you to group template options and avoid the selection of conflicting option.
Display Name - Enter a name for the template option.
 - d. Option Identifier - Enter an alphanumeric, upper case identifier for the template option. Identifiers map a template option to a corresponding template option when switching themes.
 - e. CSS Classes - Enter one or more CSS classes that define this template option.
 - f. Help Text - Enter a description of the template option.
 - g. Click **Create** or **Create and Add Another**.

Editing Global Template Options

To edit a global template option:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Themes**.The Themes page appears.
2. Select a theme.
3. Locate **Global Template Options**.
4. To edit a template option:
 - a. Click the display name.
 - b. Edit the appropriate attributes. To learn more about an attribute, see field-level Help.

Tip:

If you edit the name of a CSS class, all components that reference that CSS class name must be updated manually. The Option Identifier cannot be modified since it is needed for mapping purposes.

- c. Click **Apply Changes**.
5. To delete a template option:
 - a. Click the display name.
 - b. Click **Delete**.

Managing Template Options

This section describes how to create and edit template options.

- [Creating Template Options](#)
- [Editing Template Options](#)
- [Using Quick Edit to Modify Live Template Options](#)



See Also:

["Understanding Template Options"](#)

Creating Template Options

To create a template option:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Templates**.The Templates page appears.
2. Select a template.
3. Select **Template Options**.
4. Click **Add Template Option**.
5. Edit the appropriate attributes. To learn more about an attribute, see field-level Help.
6. Click **Create** or **Create and Add Another**.

Editing Template Options

To edit a template option:

1. Navigate to the Themes page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under User Interface, select **Templates**.

The Templates page appears.

2. Select a template.
3. Select **Template Options**.
4. Edit the appropriate attributes and click **Apply Changes**.

To learn more about an attribute, see field-level Help.

 **Tip:**

If you edit the name of a CSS class, all components that reference that CSS class name must be updated manually. The Option Identifier cannot be modified since it is needed for mapping purposes.

5. To delete a template option, click **Delete**.
6. To view global template options, expand **Global Template Options**.


Using Quick Edit to Modify Live Template Options

Developers can also use Quick Edit on the Runtime Developer toolbar to access the Live Template Options dialog. Using the Live Template Options dialog, developers can alter component's template options without refreshing the page.

To access the Live Template Options:

1. Run the page.
2. From the Runtime Developer toolbar, click **Quick Edit**.
3. Move the mouse over the component for which you want to modify template options. Click the **Wrench** icon in the upper right corner.

Sample Database Application - Customer Details

* First Name 

* Last Name

Street Address

Line 2

City

* State

* Zip Code

* Credit Limit

Phone Number

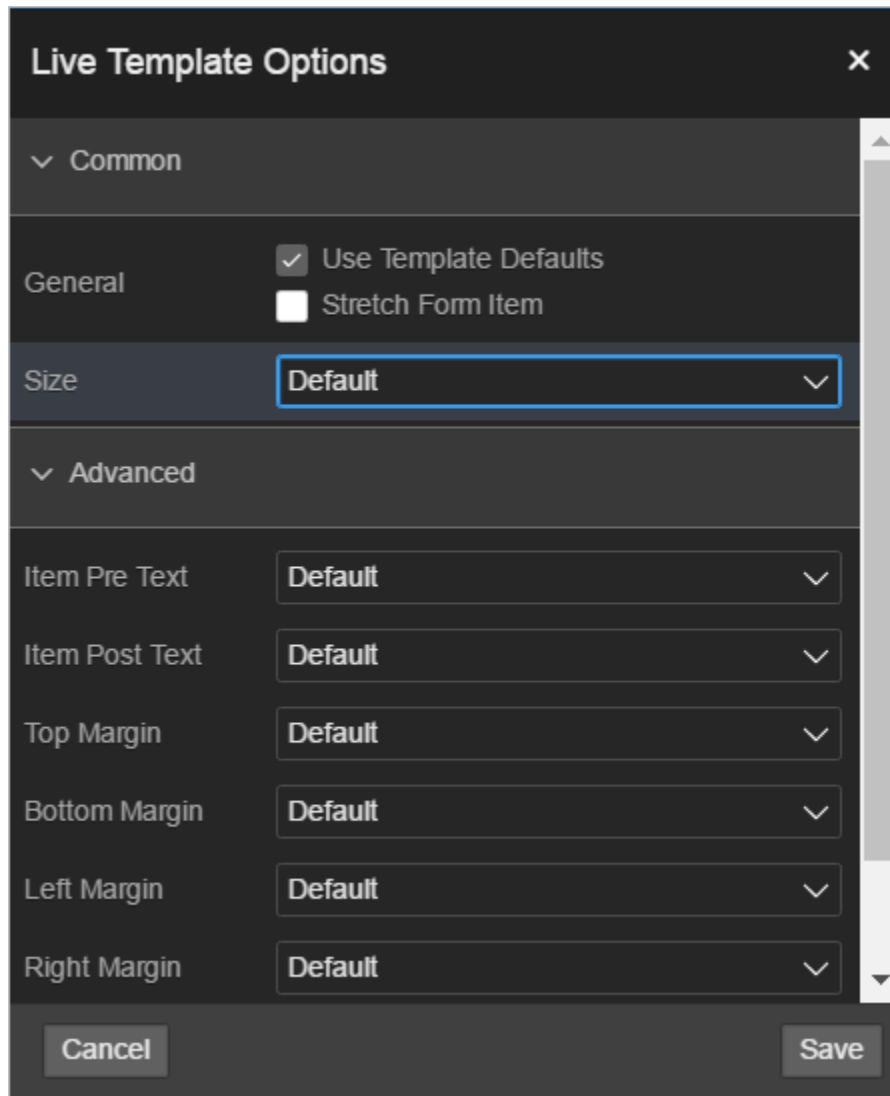
Alternate Number

Email

URL

Tags

The Live Template Options dialog appears. For region-based components, such as lists and classic reports, the dialog includes an Attributes tab.



4. Select the appropriate tab, edit the appropriate attributes, and click **Save**.



See Also:

["Runtime Developer Toolbar"](#)

Viewing Template Reports

App Builder includes reports describing template utilization, subscriptions, published templates, and edit history.

To view template reports for the current application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

3. Click **Shared Components**.
4. Under User Interface, select **Templates**.
The Templates page displays as an interactive report. To customize the view, use the Search bar at the top of the page.
5. To view template reports, click the following tabs:
 - **Subscription** - Displays subscribed templates in your application.
 - **Publish** - displays templates that have been subscribed to by other templates in your workspace. To publish master template content to subscribing templates, select those you want to publish and click Publish Checked.
 - **Utilization** - Displays template utilization in the current application for all template types. The number in the References column indicates the total number of pages that use the specific template.
 - **History** - Displays recent modifications made to Page and Region Templates in this application.

Managing Templates

You can view all available templates on the Templates page. Alternatively, you can access a template associated with a specific page.

- [Viewing Templates on the Templates Page](#)
- [Viewing Templates Associated with a Specific Page](#)
- [Replacing Templates](#)
- [Replacing All Templates within an Application](#)
- [Viewing Region Position Utilization by Page Template](#)
- [Unsubscribing to Templates](#)
- [Publishing Templates](#)



See Also:

["Viewing All Templates in a Theme"](#)

Viewing Templates on the Templates Page

To view existing templates:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.
The Templates page displays as an interactive report. To customize the view, use the Search bar at the top of the page.
5. To view or edit a template definition, click the template name.

The template definition appears.

6. Edit the appropriate attributes. To learn more about an attribute, see field-level Help.

If you edit a template, you can make changes in one window and run your application in another by selecting the **Return to Page** check box on the right side of the template definition page. Selecting this check box keeps the page you are editing current after you click Apply Changes.

7. Click **Apply Changes**.

Viewing Templates Associated with a Specific Page

To view templates associated with a specific page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. View the page in Page Designer:
 - a. In the Rendering tab, click the page title.
 - b. In the Property Editor, locate **Appearance. Page Template** displays the current template.
 - c. To save any changes, click **Save**.



See Also:

["Viewing a Page in Page Designer"](#)

Replacing Templates

You can update the template associated with a component using the Replace Templates Wizard.

To replace the template associated with a component:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.

To customize the view, use the Search bar at the top of the page.
5. From the Task list, select **Replace Templates**.
6. For Template Type:
 - a. User Interface- Select user interface.
 - b. Template Type - Identifies the template type to be replaced.
 - c. Click **Next**.
7. For 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**.
8. Click **Finish**.

Replacing All Templates within an Application

To replace all templates within an application with templates from another application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.
To customize the view, use the Search bar at the top of the page.
5. From the Task list, select **Replace templates in this application with templates from another application**.
6. For Replace Templates:
 - a. Replace from Application - Select an application.
 - b. User Interface - Select the type of user interface for which this application has been developed.
 - c. Click **Next**.
7. On Replace Templates, select a New Template for each component and then select an Action:
 - a. **Replace** copies the template definition
 - b. **Replace/Subscribe** copies the templates and adds a subscription.
8. Click **Replace Templates**.

Viewing Region Position Utilization by Page Template

Regions are organized on a page by position (or Display Point). The possible display points for a region are determined by the page-level template.

To view region position utilization by page template:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.
To customize the view, use the Search bar at the top of the page.
5. From the Task list, select **View page template region position utilization**.
The Region Display Point Utilization report appears.

Unsubscribing to Templates

A subscribed template is a template that has its definition maintained in another template, the referenced template. If your application uses subscribed templates, you can unsubscribe to templates on the Unsubscribe Templates page.

To unsubscribe to templates:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.

To customize the view, use the Search bar at the top of the page.

5. From the Task list, select **Unsubscribe Templates**.

The Unsubscribe Templates page appears.

6. Select a theme.
7. To unsubscribe to a specific template type within the current theme, select the template type and click **Unsubscribe**.

Publishing Templates

Use the Publish Templates page to view templates that are subscribed to by other templates in your workspace.

To publish master template content to subscribing templates:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.

To customize the view, use the Search bar at the top of the page.

5. Click the **Publish** tab.

The Published Templates page appears.

6. Select those you want to publish and click **Publish Checked**.

Breadcrumb Templates

A breadcrumb template controls the display of breadcrumb entries. You select a breadcrumb template when you create a region.



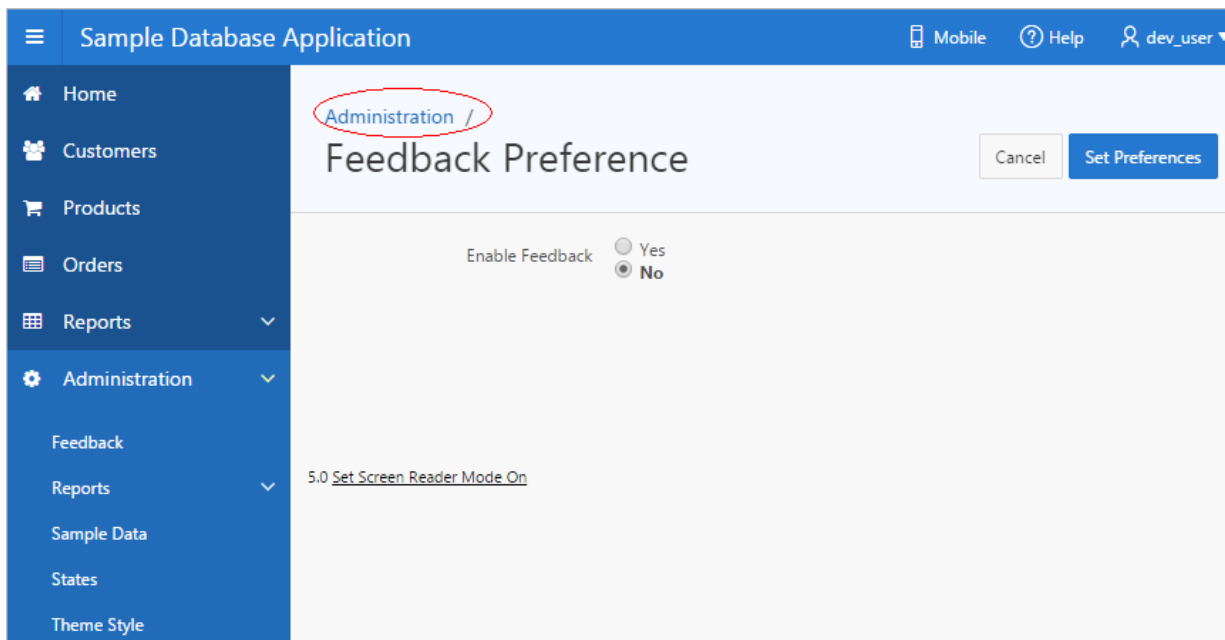
See also:

["Using Themes"](#)

- [About Breadcrumb Style Navigation](#)
- [Breadcrumb Template Attributes](#)

About Breadcrumb Style Navigation

Breadcrumbs usually indicate where the current page is relative to other pages in the application. In addition, users can click a specific page to instantly view it. For example, the Oracle Application Express Sample Database Application includes breadcrumb menus at the top of Administration pages.



See Also:

["Creating Breadcrumbs"](#)

Breadcrumb Template Attributes

A breadcrumb template controls the display of breadcrumb region types. You select a breadcrumb template when you create a region. To learn more about a specific attribute, see field-level Help.

- [Name](#)
- [Definition](#)
- [Substitution Strings](#)

Name

Name identifies the name of the template. Use the **Translatable** check box to indicate that the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member. **Template Class** identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Definition

Definition attributes control how a breadcrumb displays [Table 12-8](#) describes Definition attributes.

Table 12-8 Definition attributes

| Attribute | Description |
|-----------------------------------|--|
| Before First | Defines text that displays before the first breadcrumb entry. Use the substitution string <code>#COMPONENT_CSS_CLASSES#</code> to apply a CSS class to your breadcrumb template. You edit component CSS classes by editing the region attributes. See Also: "Editing Region Attributes " |
| Current Page Breadcrumb Entry | Defines the look of a breadcrumb entry that corresponds to the current page. |
| Non Current Page Breadcrumb Entry | Defines the look of a breadcrumb entry that does not correspond to the current page. |
| After Last | Defines text that displays after the last breadcrumb entry. |
| Between Level | Defines text that displays between levels of breadcrumb entries. |
| Maximum Levels | Specifies the number of levels that appear when displaying breadcrumbs in a breadcrumb style. |

Substitution Strings

Substitution strings are used within subtemplates to reference component values. [Table 12-9](#) describes the available button template substitution strings.

Tip:

All template substitution strings must be in uppercase letters and begin and end with a number sign (#). To view a report of substitution strings supported by a given template, see the Substitution Strings section of the Edit Breadcrumb Template page.

Table 12-9 Breadcrumb Template Substitution Strings

| Referenced From | Substitution String | Description |
|-------------------|-------------------------|--|
| Before First | #COMPONENT_CSS_CLASSES# | Component CSS classes. The #COMPONENT_CSS_CLASSES# string must be included in the before-first attribute for templates that have template options defined. The #COMPONENT_CSS_CLASSES# string is where the corresponding template option classes are injected at runtime. |
| Current Entry | #LINK# | URL Link. |
| Current Entry | #LONG_NAME# | Breadcrumb Long Name. |
| Current Entry | #NAME# | Breadcrumb Label. |
| Current Entry | #NAME_ESC_SC# | Breadcrumb Label (escaping HTML characters). |
| Non Current Entry | #LINK# | URL Link. |
| Non Current Entry | #LONG_NAME# | Breadcrumb Long Name. |
| Non Current Entry | #NAME# | Breadcrumb Label. |
| Non Current Entry | #NAME_ESC_SC# | Breadcrumb Label (escaping HTML characters). |

Button Templates

Button templates enable developers to customize the look and feel of a button. To build a button, you can use multiple images or HTML tags. Using button templates is optional.

 **See also:**
"Using Themes"

- [Button Template Attributes](#)

Button Template Attributes

This section describes some sections of the Edit Button Template page. To learn more about a specific attribute, see field-level Help.

- [Name](#)
- [Definition](#)
- [Substitution Strings](#)

**See Also:**

"Understanding Template Options" and "Understanding Theme Subscriptions"

Name

Template Name identifies the name of the template. Use the **Translatable** check box to indicate if the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Definition

Normal Template defines the button template that displays if the button type is **Normal**. For example, `&ITEM_NAME.` values can be substituted at rendering time. **Hot Template** defines the button template that displays if the button type is **Hot**. You have the option of including standard application substitutions. For example, `&ITEM_NAME.` values can be substituted at rendering time.

To learn more about supported substitution strings, see field-level Help.

**Tip:**

All substitution string must be in uppercase.

Substitution Strings

Substitution strings are used within subtemplates to reference component values. [Table 12-10](#) describes the available button template substitution strings.

**Tip:**

All template substitution strings must be in uppercase letters and begin and end with a number sign (#). To view a report of substitution strings supported by a given template, see the Substitution Stings section of the Edit Page Template page.

Table 12-10 Button Template Substitution Strings

| Referenced From | Substitution String | Description |
|-----------------|---------------------|--------------------|
| Hot Template | #BUTTON_ATTRIBUTES# | Button attributes. |

Table 12-10 (Cont.) Button Template Substitution Strings

| Referenced From | Substitution String | Description |
|-----------------|----------------------|---|
| Hot Template | #BUTTON_CSS_CLASSES# | Button CSS classes. You must include the #BUTTON_CSS_CLASSES# string for templates that have template options defined. The #BUTTON_CSS_CLASSES# string is where the corresponding template option classes are injected at runtime. |
| Hot Template | #BUTTON_ID# | Generated button ID will be either the button's Static ID if defined, or if not will be an internally generated ID in the format 'B' [Internal Button ID]. |
| Hot Template | #ICON_CSS_CLASSES# | Icon CSS classes. |
| Hot Template | #JAVASCRIPT# | Used in an onclick attribute. |
| Hot Template | #LABEL# | Button label. |
| Hot Template | #LINK# | To be used in a href attribute. |
| Template | #BUTTON_ATTRIBUTES# | Button attributes. |
| Template | #BUTTON_CSS_CLASSES# | Button CSS classes. You must include the #BUTTON_CSS_CLASSES# string for templates that have template options defined. The #BUTTON_CSS_CLASSES# string is where the corresponding template option classes are injected at runtime. |
| Template | #BUTTON_ID# | Generated button ID will be either the button's Static ID if defined, or if not will be an internally generated ID in the format 'B' [Internal Button ID]. |
| Template | #ICON_CSS_CLASSES# | Icon CSS classes. |
| Template | #JAVASCRIPT# | Used in an onclick attribute. |
| Template | #LABEL# | Button label. |
| Template | #LINK# | To be used in a href attribute. |

Legacy Calendar Templates

Legacy Calendar templates control the appearance and placement of a calendar. Calendar templates frequently use HTML tables to arrange dates. You place calendar attributes using substitution strings such as #DD# and #MONTH#.

- [Supported Legacy Calendar Template Substitution Strings](#)
- [Legacy Calendar Template Attributes](#)



See Also:

"About Supported Calendars," "Creating Calendars," and "Using Themes"

Supported Legacy Calendar Template Substitution Strings

A list of supported substitution strings appears on the right side of the Edit Calendar Template page. Note that template substitution strings must be in uppercase letters and begin and end with a number sign (#).

Legacy Calendar Template Attributes

This section describes some sections of the Edit Calendar Template page. To learn more about a specific attribute, see field-level Help.



See Also:

["Understanding Template Options"](#) and ["Understanding Theme Subscriptions"](#)

Calendar Definition

- [Name](#)
- [Subscription](#)
- [Comments](#)

Name

Name identifies the name of the template. Use the **Translatable** check box to indicate if the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Subscription

Use Subscription to apply an existing template to the current application.

Use **Reference Master Label Template From** to select an existing template in this workspace or another template in your workspace's schema. By selecting an existing template, you become a subscriber to that template.

To load a copy of a master template, click **Refresh**.

Comments

Use this attribute to record comments about this component.

Monthly Calendar, Weekly Calendar, Daily Calendar, Custom Calendar and List Calendar

Use the Monthly Calendar, Weekly Calendar, Daily Calendar, and List Calendar attributes to control the appearance and placement of specific calendars.

Label Templates


Label templates are designed to centrally manage HTML markup of page item labels. Each item can have an optional label. You can control how these labels display using label templates. For example, you could create a label template called Required Field that references an image (such as an asterisk) to indicate to the user that the field is required.

Label templates enable you to define a before-and-after text string that gets prepended and appended to the item.

- [Label Template Attributes](#)
- [About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help](#)

 **See Also:**
"Using Themes"

Label Template Attributes

 **See Also:**
"Understanding Template Options" and "Understanding Theme Subscriptions"

This section describes specific sections of the Edit Label Template page. To learn more about a specific attribute, see field-level Help.

- [Name](#)
- [Definition](#)
- [Error Display](#)
- [Field Container](#)
- [Substitution Strings](#)

Name

Template Name identifies the name of the template. Use the **Translatable** check box to indicate that the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Definition

Definition attributes include:

Tip:

For a list of supported substitution strings and to view examples, see field-level Help.

- **Before Label** - Enter HTML to display before the item label. The `#ITEM_CSS_CLASSES#` substitution string must be included for templates that have template options defined.
- **After Label** - Enter HTML to display after the item label. Since the label automatically displays before the HTML in this region, any open HTML tags in the Before Label region should be closed.
- **Before Item** - Enter HTML to be displayed directly before the form item. This is typically used to wrap an item into a DIV container or to display a help button before or after the form item.
- **After Item** - Enter HTML to be displayed directly after the form item. This is typically used to wrap an item into a DIV container or to display a help button before or after the form item.
- **Item Pre Text** - Enter HTML for the Item Pre Text subtemplate to format the content stored as Pre Text in the item attributes. This subtemplate can be referenced in the **Before Element** template with the `#ITEM_PRE_TEXT#` substitution.
- **Item Post Text** - Enter HTML for the Item Post Text subtemplate to format the content stored as Post Text in the item attributes. This subtemplate can then be referenced in the **After Element** template with the `#ITEM_POST_TEXT#` substitution.
- **Help Template** - The help sub template enables developers to define the link or button used to invoke the item level help. This sub template is only rendered for items that have help defined. See "[About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help.](#)"
- **Inline Help Template** - The help sub template enables developers to define inline help text to be displayed immediately with the item. This sub template is only rendered for items that have inline help text defined.

Error Display

Error Display attributes include:

 **Tip:**

For a list of supported substitution strings and to view examples, see field-level Help.

- **On Error Before Label** - Enter HTML to precede the item label when an application displays an inline validation error message for the item.
- **On Error After Label** - Enter HTML to be appended to the item label when an application displays an inline validation error message for the item.
- **Error Template** - Enter the HTML rendered to display the inline validation error message. Use the `#ERROR_MESSAGE#` substitution string to reference the message text. The error template can be included in the before and after item attributes using the `#ERROR_TEMPLATE#` substitution string.

Field Container

Field Container attributes include:

 **Tip:**

For a list of supported substitution strings and to view examples, see field-level Help.

- **Before Label and Item** - Enter HTML to be displayed before the label and item. This is typically used to wrap an item into a DIV container or field set. The Before Label and Item attribute is only rendered for pages that do not use a table grid to display form items. This includes any page using a mobile page template.
- **After Label and Item** - Enter HTML to be displayed after the label and item. This is typically used to wrap an item into a DIV container or field set. The after item and label attribute is only rendered for pages that don't use a table grid to display form items. This includes any page using a mobile page template.

Substitution Strings

Lists substitution string usage for this template. Substitution strings are used within subtemplates to reference component values.

About Using `#CURRENT_ITEM_HELP_TEXT#` to Create Item Help

You can use the `#CURRENT_ITEM_HELP_TEXT#` substitution string to create inline page Help text when a page is rendered. For example, you can use this substitution string to create page Help text which displays as a tooltip when the mouse hovers over the page item label. This placeholder enables you to include help text on the page when it is rendered without having to perform an expensive Ajax call.

**Tip:**

Only use this placeholder if the Help text is short. The generated page always includes the help text of all page items.

Example Label Template

Before Label:

```
<label for="#CURRENT_ITEM_NAME#" id="#LABEL_ID#">  
<span class="helpText">#CURRENT_ITEM_HELP_TEXT#</span>
```

After Label:

```
</label>
```

List Templates

A list is a shared collection of links. You control the appearance of a list through list templates. Using template attributes, you can also define a list element to be either current or non current for a specific page.

- [About Hierarchical Lists](#)
- [List Template Attributes](#)

**See Also:**

["Creating Lists"](#) and ["Using Themes"](#)

About Hierarchical Lists

Oracle Application Express supports hierarchical lists. To create a hierarchical list, you must:

- Select a list template that supports hierarchical lists. To determine which list templates support hierarchical lists, look for templates having the naming convention "with Sublist."
- Select a Parent List Entry when you create each list entry.

**See Also:**

["Creating Lists"](#)

List Template Attributes



See Also:

"Understanding Template Options" and "Understanding Theme Subscriptions"

This section describes some sections of the Edit List Template page. To learn more about a specific attribute, see field-level Help.

- [Name](#)
- [Before List Entry](#)
- [Template Definition](#)
- [Template Definitions for First Entry](#)
- [Before Sublist Entry](#)
- [Sublist Entry](#)
- [After Sub List Entry](#)
- [After List Entry](#)
- [Substitution Strings](#)

Name

Name identifies the name of the template. Use the **Translatable** check box to indicate that the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Before List Entry

Enter HTML that displays before any list elements. You can use this attribute to open an HTML table or HTML table row. The `#COMPONENT_CSS_CLASSES#` substitution string must be included in the List Template Before Rows attribute for templates that have template options defined.

Template Definition

Defines current and noncurrent list templates. Supported substitution strings include `#LINK#`, `#TEXT#`, `#TEXT_ESC_SC#`, `#ICON_CSS_CLASSES#`, `#IMAGE_PREFIX#`, `#IMAGE#`, `#IMAGE_ATTR#`, and `#A01#...#A10#`.

Template Definition attributes include:

- **List Template Current.** Enter HTML or text to be substituted for the selected (or current) list template.

- **List Template Current with Sub List Items.** Enter HTML or text to be substituted for the selected (or current) list template when an item has sublist items. If not specified, the current list item template is used.
- **List Template Noncurrent.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template.
- **List Template Noncurrent with Sub List Items.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template used when an item has sublist items. If not specified, the current list item template is used.
- **Between List Elements.** Enter HTML that displays between list elements. This attribute is ignored if no HTML is specified.

Template Definitions for First Entry

Available attributes include:

- **List Template Current (First).** If defined, this is used for the first list item. Enter HTML or text to be substituted for the selected (or current) list template.
- **List Template Current with Sub List Items (First).** If defined, this is used for the first list item. HTML or text to be substituted for the selected (or current) list template used when item has sub list items.
- **List Template Noncurrent (First).** If defined, this is used for the first list item. Enter HTML or text to be substituted for the unselected (or noncurrent) list template.
- **List Template Noncurrent with Sub List Items (First).** If defined, this is used for the first list item. HTML or text to be substituted for the unselected (or noncurrent) list template used when item has sub list items.

Before Sublist Entry

Enter HTML that displays before any sublist elements.

Sublist Entry

Defines current and noncurrent list templates. Supported substitution strings include #LINK#, #TEXT#, #ICON_CSS_CLASSES#, #IMAGE_PREFIX#, #IMAGE#, #IMAGE_ATTR#, #ICON_CSS_CLASSES#, #A01#...#A10#, #LIST_ITEM_ID#, and #PARENT_LIST_ITEM_ID#.

Sub List Entry attributes include:

- **Sub List Template Current.** Enter HTML or text to be substituted for the selected (or current) list template.
- **Sub List Template Current with Sub List Items.** Enter HTML or text to be substituted for the selected (or current) list template when an item has sublist items. If not specified, the current list item template is used.
- **Sub List Template Noncurrent.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template.
- **Sub List Template Noncurrent with Sub List Items.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template used when an item has sublist items. If not specified, the current list item template is used.
- **Between Sub List Items.** Enter HTML that displays between list elements. This attribute is ignored if no HTML is specified.

After Sub List Entry

Enter HTML that displays after displaying sublist elements.

After List Entry

Enter HTML that displays after displaying all list elements. You can use this attribute to close an HTML table opened in the Before List Elements attribute.

Substitution Strings

Lists substitution string usage for this template. Substitution strings are used within subtemplates to reference component values.

Page Templates


Page templates define the appearance of a page. Each template consists of a header template, a body template, a footer template, and several subtemplates. If you do not specify a page template as a page-level attribute, then the Application Express engine uses the default page template defined on the Create/Edit Theme page.

Page templates combine static HTML with substitution strings that are replaced at runtime. You use substitution strings to indicate the existence and placement of a component within a page template. You can further specify how a component should display using subtemplates.

- [Page Template Attributes](#)
- [Required Dialog Page Template Attributes](#)

 **See Also:**
["Using Themes"](#)

Page Template Attributes

 **See Also:**
["Understanding Template Options"](#) and ["Understanding Theme Subscriptions"](#)

This section describes specific sections of the Edit Page Template page. To learn more about a specific attribute, such as supported substitution strings, see field-level Help.

- [Name](#)
- [Definition](#)

- [JavaScript](#)
- [Cascading Style Sheet](#)
- [Subtemplate](#)
- [Image Based Tab Attributes](#)
- [Layout](#)
- [Display Points](#)
- [Dialog](#)
- [Error Page Template Control](#)
- [Substitution Strings](#)

Name

Name identifies the name of the template. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Use the **Translatable** check box to indicate if this template contains text strings that require translation.

Template Type identifies the page template type. If set to **Dialog Page**, the template is only be available for selection by a page with a page mode of **Modal** or **Non-Modal**. If set to **Normal Page**, the template is only be available for selection by a page with a page mode of **Normal**.



See Also:

["Creating Dialog Pages"](#) and ["Required Dialog Page Template Attributes"](#)

Definition

Each template consists of a header, a body, a footer, and subtemplates. Use substitution strings to include dynamic content. All template substitution strings must be in uppercase letters and begin and end with a number sign (#).



Tip:

For a list of supported substitution strings and to view examples, see field-level Help.

Definition attributes include:

- **Header** - Enter HTML that defines the <Head> section of the HTML document. Oracle recommends you include at least #TITLE# and #HEAD# in this template section.
- **Body** - Enter HTML that makes up the <Body> section of the HTML document. Include substitution strings to define where the body content is placed and to define additional positions for regions.
- **Footer** - Enter HTML to define third section in the page template that displays after the Body.

JavaScript

Available attributes include:

Tip:

For a list of supported substitution strings and to view examples, see field-level Help.

- **File URLs** - Enter JavaScript file URLs for code to be loaded on this page. Each URL must be written into a new line. If you provide a minified version of your file you can use the substitution string #MIN# to include .min in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You can also use the substitution string #APP_VERSION# to include the application's version in the file URL. JavaScript file URLs you enter here replaces the #TEMPLATE_JAVASCRIPT# substitution string in the page template.
Do not include opening or closing script tags, just write the URL.
- **Function and Global Variable Declaration** - Enter JavaScript code (for example, functions or global variable declarations) for code to be used in this page template. Consider putting it into an external file to avoid duplication and to leverage browser caching of static files. Code you enter here replaces the #TEMPLATE_JAVASCRIPT# substitution string in the page template.
Do not include opening or closing script tags, just include the JavaScript code.
To reference a shortcut, use "SHORTCUTNAME".
- **Execute when Page Loads** - Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle Application Express. Code you enter here replaces the #GENERATED_JAVASCRIPT# substitution string in the page template.
- **Dialog Initialization Code** - This attribute only displays if the Template Type is **Dialog Page**. Enter JavaScript code to initialize the page as a dialog. The substitution strings will be replaced with their respective values at rendering time.
- **Dialog Closure Code**. This attribute only displays if the Template Type is **Dialog Page**. Enter JavaScript code to close the dialog page.
- **Dialog Cancel Code**. This attribute only applies if the Template Type is **Dialog Page**. Enter JavaScript code to cancel the dialog page.

Cascading Style Sheet

Available attributes include:

- **File URLs** - Enter Cascading Style Sheet file URLs to be loaded for this page template. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You can also use the substitution string `#APP_VERSION#` to include the application's version in the file URL. File URLs you enter here replace the `#TEMPLATE_CSS#` substitution string in the template.

Do not include opening or closing script tags, just include the file URL.

- **Inline** - Enter Cascading Style Sheet code to be used for this page template. Consider putting it into an external file to avoid duplication and to leverage browser caching of static files. Code you enter here replaces the `#TEMPLATE_CSS#` substitution string in the template.

Do not include opening or closing style tags, just include the cascading style sheet code.

Subtemplate

Enter the HTML for subtemplates below. Subtemplates are placed inside the body of the page template using substitution strings.

Subtemplates include:

- **Success Message** - Enter HTML to replace the string `#SUCCESS_MESSAGE#` in the template body, header, or footer.
- **Navigation Bar** - Enter HTML or text to replace the string `#NAVIGATION_BAR#` in the template header, body, or footer. Use the substitution string `#BAR_BODY#` to indicate where in the Navigation Bar the entries should appear.
- **Navigation Bar Entry** - Enter HTML or text to be substituted into the navigation bar `#BAR_BODY#` substitution string for each navigation bar entry.
- **Notification** - Enter HTML or text to be substituted when the `#NOTIFICATION_MESSAGE#` substitution string is referenced in the template header, body, or footer.

Image Based Tab Attributes

Use this subtemplate for tabs that are entirely based on images. Available attributes include:

- **Current Image Tab** - Enter HTML to be used to indicate that an image-based tab is currently selected. Include the `#TAB_TEXT#` substitution string to show the displayed name of the tab.
- **Non Current Image Tab** - Enter the HTML to be used to indicate that an image tab is not currently selected. Include the `#TAB_TEXT#` substitution string to show the displayed name of the tab.

Layout

Specifies the type of layout used for the page. The Type you select determines what attributes displays. To learn more about an attribute, see field-level Help.

From Type, select a layout:

- **HTML Table** - Uses HTML Tables to position regions, page items, and buttons.
- **Fixed Number of Columns** - Used for grids which divide the screen into a fixed number of grid columns.
- **Variable Number of Columns** - Used for grids which can display up to "n" number of grid columns depending on how the regions, page items and buttons are positioned.

Display Points

Available attributes include:

- **Breadcrumb Display Point** - Applies to generated components that use breadcrumbs. It defines where the breadcrumbs are placed on the page. Sidebar Display Point
- **Sidebar Display Point** - Applies to generated components that use Sidebars. It defines where sidebars are placed on the page.

Dialog

Dialog attributes only appear if you select a Template Type of **Dialog Page**. These settings are overwritten by a similar attributes specified at page-level.

Available Dialog attributes include:

- **Width** - Enter the width of the dialog, in pixels or a percentage.
- **Height** - Enter the height of the dialog in pixels, for example, 500.
- **Maximum Width** - Enter the maximum width of the dialog, in pixels. For a page using the jQuery Mobile Smartphone user interface, the maximum width is 500.
- **CSS Classes** - Enter additional CSS classes you wish to be applied to the dialog, as a space delimited list.
- **Allow Embed in Frames** - Your page template level browser frame options are listed below, along with the current setting for each. Make a selection to control if the browser is allowed to display your application's pages within a frame. The current setting will be the default for each modal and non-modal page using this page template. To learn more, see field-level Help.

Oracle Application Express embeds modal dialogs in iframes. If you set the application security attribute Browser Frame to **Deny**, checking **Allow Embed In Frame** for Modal Dialog will instead use **Allow from same origin** for dialog pages that use this template

**See Also:**["Required Dialog Page Template Attributes"](#)

Error Page Template Control

Use this attribute when a page template is designated as an error template.

Substitution Strings

[Table 12-11](#) describes the available page template substitution strings. Substitution strings are used within subtemplates to reference component values.

**Tip:**

All template substitution strings must be in uppercase letters and begin and end with a number sign (#). To view a report of substitution strings supported by a given template, see the Substitution Strings section of the Edit Page Template page.

Table 12-11 Page Template Substitution Strings

| Referenced From | Substitution String | Description |
|-----------------|-------------------------------|--|
| All | #HIGH_CONTRAST_TOGGLE# | High Contrast toggle. |
| All | #PAGE_STATIC_ID# | Page Static ID (for example 'P100') |
| All | #PARENT_TAB_CELLS# | Display of parent tabs. |
| All | #REGION_POSITION_01# | Region Position 1. |
| All | #REGION_POSITION_02# | Region Position 2. |
| All | #REGION_POSITION_03# | Region Position 3. |
| All | #REGION_POSITION_04# | Region Position 4. |
| All | #REGION_POSITION_05# | Region Position 5. |
| All | #REGION_POSITION_06# | Region Position 6. |
| All | #REGION_POSITION_07# | Region Position 7. |
| All | #REGION_POSITION_08# | Region Position 8. |
| All | #REQUESTED_URL# | Page URL requested by the browser |
| All | #SCREEN_READER_TOGGLE# | Screen Reader toggle. |
| All | #SWITCH_UI_TO_DESKTOP# | Hyperlink for home page of desktop user interface. |
| All | #SWITCH_UI_TO_JQM_SMARTPHONE# | Hyperlink for home page of smart phone user interface. |
| All | #TITLE# | Defines the page title. Typically included within HTML title tags. |
| Body | #BODY# | Page Body. |

Table 12-11 (Cont.) Page Template Substitution Strings

| Referenced From | Substitution String | Description |
|-----------------|------------------------|---|
| Body | #BOX_BODY# | Where the Body displays. If the Body is null, then #BOX_BODY# is used instead. |
| Body | #FORM_OPEN# | Where the HTML open form tag <form> is placed. You must include this substitution string to submit a form. |
| Body | #GENERATED_CSS# | Cascading style sheet files created by components. |
| Body | #GLOBAL_NOTIFICATION# | Displays the Global Notification attribute. Global notifications are intended to communicate system status, such as pending system downtime. You can also use APEX_APPLICATION.G_GLOBAL_NOTIFICATION to set this value programmatically. See Also: "Global Notification" |
| Body | #LOGO# | Application image or log. In the Logo section of the Edit Application Definition page, you can identify an image and image attributes for an application logo. To use this feature, you must also include the #LOGO# substitution string in the Header or Body page template. See Also: "Logo" |
| Body | #NAVIGATION_BAR# | Navigation bar. See Also: "Subtemplate" for information about the Navigation Bar subtemplate |
| Body | #NOTIFICATION_MESSAGE# | Defines where a summary of inline error messages displays. Inline error messages can display next to a field, inline in the notification area, or both. |
| Body | #POPUP_TRANSITION# | Default page transition used to open a dialog in a jQuery Mobile application. You must use this new placeholders in a DIV of type role="page". For example: <pre><div id="#PAGE_STATIC_ID#" data-role="page" data-apex-page-transition="#PAGE_TRANSITION#" data-apex-popup-transition="#POPUP_TRANSITION#" data-url="#REQUESTED_URL#"></pre> |
| Body | #SUCCESS_MESSAGE# | Success message. Defines where page success and error messages appear. If the page process runs without raising errors, then this text displays. You can customize the display of the success message for each template by adding HTML to be displayed before and after the success message. |
| Body | #TAB_CELLS# | Display of standard tabs. |
| Body | #TEMPLATE_JAVASCRIPT# | Display of standard tabs. |
| Footer | #CUSTOMIZE# | Customize link. |
| Footer | #FORM_CLOSE# | Where the HTML close form tag <form> is placed. |

Table 12-11 (Cont.) Page Template Substitution Strings

| Referenced From | Substitution String | Description |
|-----------------|------------------------------|---|
| Footer | #BUILT_WITH_LOVE_USING_APEX# | Displays "Built with ♥ using Oracle APEX." |
| Header | #APEX_CSS# | Cascading Style Sheet files used by Application Express. |
| Header | #APEX_JAVASCRIPT# | JavaScript files and code used by Oracle Application Express. |
| Header | #APPLICATION_CSS# | CSS files defined on user interface level. |
| Header | #APPLICATION_JAVASCRIPT# | JavaScript files defined on user interface level. |
| Header | #GENERATED_JAVASCRIPT# | JavaScript code created by components as needed. |
| Header | #HEAD# | Used after the <head> open tag but before the </head> close tag. You can optionally define the contents of #HEAD# for each page (for example, to reference additional style sheets or JavaScript libraries). |
| Header | #ONLOAD# | Page onload. Use this string as a substitute in a JavaScript call to be executed when a page is loaded by the web browser. The JavaScript to be called can vary for each page. |
| Header | #PAGE_CSS# | Cascading style sheet files and inline style defined on the page-level. |
| Header | #PAGE_JAVASCRIPT# | JavaScript files and code defined on the page-level. |
| Header | #TEMPLATE_CSS# | Cascading style sheet files and inline style defined on page template-level. |
| Header | #THEME_CSS# | Cascading Style Sheet files defined at the theme-level. |
| Header | #THEME_JAVASCRIPT# | JavaScript files defined on theme-level. |
| Header | #THEME_STYLE_CSS# | Cascading Style Sheet files defined on the theme style level. |
| Page | #FAVICON# | Enables support for the application User Interface attribute, Favicon. Developers can use the Favicon attribute for creating a favicon (or shortcut icon). To use this feature, your page template must include the #FAVICON# substitution string. |
| Page | #PAGE_TRANSITION# | Default page transition used to navigate to other pages in a jQuery Mobile application. You must use this new placeholders in a DIV of type role="page". For example: <pre><div id="#PAGE_STATIC_ID#" data-role="page" data-apex-page-transition="#PAGE_TRANSITION#" data-apex-popup-transition="#POPUP_TRANSITION#" data-url="#REQUESTED_URL#"></pre> |
| Page | #PAGE_URL# | Page URL. |

Table 12-11 (Cont.) Page Template Substitution Strings

| Referenced From | Substitution String | Description |
|-----------------|-------------------------------|---|
| JavaScript | #DIALOG_ATTRIBUTES# | Defines dialog attributes not declaratively supported. |
| JavaScript | #DIALOG_CSS_CLASSES# | Dialog CSS Classes. |
| JavaScript | #DIALOG_HEIGHT# | Dialog Height. |
| JavaScript | #DIALOG_MAX_WIDTH# | Dialog Maximum Width. |
| JavaScript | #DIALOG_WIDTH# | Dialog Width. |
| JavaScript | #IS_MODAL# | Dialog Page Mode (for example, true when modal and false when non-modal). |
| JavaScript | #SIDE_GLOBAL_NAVIGATION_LIST# | Navigation Menu (Side). |
| JavaScript | #TOP_GLOBAL_NAVIGATION_LIST# | Navigation Menu (Top). |
| JavaScript | #TRIGGERING_ELEMENT# | Triggering Element (for example, the current DOM element). |

Required Dialog Page Template Attributes

To create a dialog page template, you must define the Page Template attributes described in [Table 12-12](#).

Table 12-12 Required Attributes for Dialog Page Templates

| Attribute | Description |
|--|---|
| Name, Template Type | Must be defined as Dialog Page . |
| JavaScript, Dialog Initialization Code | Enter JavaScript code to initialize the page as a dialog. See field-level Help for function call syntax. |
| JavaScript, Dialog Closure Code | Enter JavaScript code to close the dialog page. See field-level Help for function call syntax. |
| JavaScript, Dialog Cancel Code | Enter JavaScript code to cancel the dialog page. See field-level Help for function call syntax. |
| Dialog, Width | Sets the default width for dialogs. The page template-level setting can be overwritten at page-level. |
| Dialog, Height | Sets the default height for dialogs. The page template-level setting can be overwritten at page-level. |
| Dialog, Maximum Width | Sets the maximum width for dialogs, in pixels. The page template-level setting can be overwritten at page-level. |
| Dialog, CSS Classes | Enter additional CSS classes you wish to be applied to dialogs. The page template-level setting can be overwritten at page-level. |
| Dialog, Allow Embed in Frame | Select the page mode to use this attribute to control if the browser is allowed to display your application's pages within a frame. Application Express embeds modal dialogs in iframes. If you set the application security attribute Browser Frame to Deny , checking Allow Embed In Frame for Modal Dialog will instead use Allow from same origin for dialog pages that use this template. |

**See Also:**

["Creating Dialog Pages"](#) and ["Page Template Attributes"](#)

Popup LOV Templates

Popup LOV templates control how popup lists display for all items defined as `POPUP`. You can only specify one popup LOV template for each theme.

- [Popup List of Values Template Attributes](#)

**See Also:**

["Creating Lists of Values at the Application-Level"](#) and ["Using Themes"](#)

Popup List of Values Template Attributes

**See Also:**

["Understanding Theme Subscriptions"](#)

This section describes some sections of the Edit Popup List of Values Template page. For more information on an attribute, see field-level Help.

- [Search Field](#)
- [Buttons](#)
- [Window](#)
- [Pagination](#)
- [Result Set](#)
- [Page Attributes](#)

Search Field

Use these attributes to specify how a Search field displays. [Table 12-13](#) describes available Search Field attributes.

Table 12-13 Search Field Attributes

| Attribute | Description |
|-------------------|--|
| Before Field Text | Defines text to display before the popup list of values search field displays. |

Table 12-13 (Cont.) Search Field Attributes

| Attribute | Description |
|-----------------------|---|
| After Field Text | Displays this text after displaying the search field, the search button, and the close button. |
| Filter Width | Displays the HTML <code>INPUT TYPE = TEXT</code> widget using this width. |
| Filter Max Width | Displays the HTML <code>INPUT TYPE = TEXT</code> widget using this maximum width. |
| Filter Text Attribute | Displays the HTML <code>INPUT TYPE = TEXT</code> widget using these attributes. This is included within the HTML input tag. |

Buttons

Use these attributes to define the button name and attributes for the Find, Close, Next, and Previous buttons.

Window

Popup lists of values are executed using JavaScript. Use these attribute to control the values of `scrollbars=`, `resizable=`, `width=`, and `height=`.

Pagination

Defines how row count results display.

Result Set

Use these attributes to define text or HTML to display before and after a result set.

Page Attributes

Use these attributes to define popup pages.

Region Templates



See Also:

["Understanding Template Options"](#) and ["Understanding Theme Subscriptions"](#)

Region templates control the appearance and placement of region attributes. You place region attributes using substitution strings such as `#BODY#` and `#EDIT#`. Template substitution strings must be in upper case and begin and end with a pound sign (`#`).

- [Region Template Attributes](#)

**See Also:**

["About Regions"](#) and ["Using Themes"](#)

Region Template Attributes

**See Also:**

["Understanding Template Options"](#) and ["Understanding Theme Subscriptions"](#)

This section describes some attributes on the Edit Region Template page. To learn more about a specific attribute, see field-level Help.

- [Name](#)
- [Definition](#)
- [Layout](#)
- [Sub Regions](#)
- [Substitution Strings](#)

Name

Theme indicates the theme to which the template is a member. **Name** identifies the name of the template.

Template Class identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class. Use the **Translatable** check box to indicate that the template contains text strings that require translation.

Definition

Region templates provide the appearance for a portion of a page called a region. Use substitution strings to indicate the existence and placement of a component within the region. #BODY# is the only required substitution string and identifies where the source of the region should be placed. All other substitution strings are optional. The following are valid substitution strings:

- #TITLE#
- #PAGE_TITLE#
- #EXPAND#
- #CHANGE#
- #BODY#
- #SUB_REGION_HEADERS#

- #SUB_REGIONS#
- #FORM_OPEN#
- #FORM_CLOSE#
- #REGION_ID#
- #REGION_STATIC_ID#
- #REGION_CSS_CLASSES#
- #REGION_ATTRIBUTES#
- #REGION_HEADER#
- #REGION_FOOTER#

When you create a button in a region position, the positions you define appear in a select list. Use the following substitution strings to define positions for the placement of buttons in a region:

- #EDIT#
- #CLOSE#
- #CREATE#
- #CREATE2#
- #EXPAND#
- #HELP#
- #DELETE#
- #COPY#
- #NEXT#
- #PREVIOUS#



See Also:

["Using Substitution Strings"](#)

Layout

Page items display within regions. If the layout in the page template is set to **Table**, each item is part of an HTML table. Use this attribute to set the attributes of this table.

Sub Regions

Use **Header Templates** and **Header Entry Templates** to generate a list of region titles of all the subregions of the current region. For either template, #ENTRIES# is the only required substitution string.

Use **Template** to wrap a subregion with additional HTML code. #SUB_REGION# is the only required substitution string. It identifies where the source of the subregion should be placed. All other substitution strings are optional.

Substitution Strings

Lists substitution string used within templates to reference component values.

Report Templates

Report column templates provide you with control over the results of a row from a SQL query. This type of template defines a cell, not an entire row.

Each report template identifies column names using the syntax #1#, #2#, #3#, and so on. You can also name columns using column name substitution syntax such as #ENAME# or #EMPNO#. You can reference any item from your application within your template. For example, to reference an item called `ABC.` in your template, you could include the exact substitution string `&ABC.`. The actual value of `ABC.` would be provided by an end user editing an item in your application named `ABC.`

- [Generic Column Templates and Named Column Templates](#)
- [Report Column Template Attributes for Generic Column Templates](#)
- [Report Column Template Attributes for Named Column Templates](#)
- [About Using JavaScript in Column Templates](#)



See Also:

["Using Themes," "Understanding Template Options," and "Understanding Theme Subscriptions"](#)

Generic Column Templates and Named Column Templates

Oracle Application Express includes two types of report templates:

- [Generic Column Templates](#)
- [Named Column Templates](#)

Generic Column Templates

A generic column template determines the appearance of a report by defining the look of the column once. This look is then repeated as many times as is necessary, based on the number of columns specified in the report's definition. This type of template is limited to reports that have a standard row and column structure. Additional style can be applied to a report using this type of template through the use of conditions.

The following example demonstrates how to have each column use a specific style:

```
<td class="tabledata" align="#ALIGN#">#COLUMN_VALUE#</td>
```

This example assumes your page template includes a CSS containing the class `tabledata`. This example also demonstrates the use the substitution strings `#ALIGN#` and `#COLUMN_VALUE#`. If you actually ran this report, these substitution strings would be replaced with values generated by the results of a SQL query.

If your query uses an expression in the select list, it is a good idea to create an alias for the columns to avoid runtime errors. For example, suppose your query was as follows:

```
SELECT ename, (sal + comm) * 12 FROM emp
```

You could rewrite the query to alias the columns as follows:

```
SELECT ename, (sal + comm) * 12 yearly_comp FROM emp
```

Named Column Templates

Named column templates allow for more flexibility in report design. However, because they reference columns by name, they can only be used by reports that are based on those columns. For example:

```
<tr><td>#ENAME#</td><td>#SAL#</td></tr>
```

Although named column templates offer flexibility, you may need to create a new template for each query. You can also include a position notation. The following example demonstrates how to use following HTML and substitution strings:

```
<tr><td>#ENAME#</td><td>#SAL#</td></tr>
```

```
<tr><td>#1#</td><td>#2#</td></tr>
```

Report Column Template Attributes for Generic Column Templates

This section describes specific sections of the Edit Report Template page for Generic Column Templates. To learn more about a specific attribute, see field-level Help.

- [Report Template](#)
- [Before Rows](#)
- [Column Headings](#)
- [Before Each Row](#)
- [Column Templates](#)
- [After Each Row](#)
- [After Rows](#)
- [Row Highlighting](#)
- [Pagination Subtemplate](#)
- [Comments](#)

Report Template

Template Name identifies the name of the template. **Template Type** indicates the type of template. Named Column templates reference column names in the template. Generic Column Templates reference the #COLUMN_VALUE# substitution string in the template.

Theme indicates the theme to which the template is a member. **Template Class** identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template

class. Use the **Translatable** check box to indicate the template contains text strings that require translation.

Before Rows

In **Before Rows**, enter HTML that displays once at the beginning of a report template.

Opening an HTML table is a common use of this attribute as shown in the following example:

```
<table>
```

You can include pagination above a report by including the substitution string `#TOP_PAGINATION#`. This substitution string generates HTML that starts with an opening `<tr>` tag and ends with a closing `</tr>` tag. For example, to include an open table tag and the `#TOP_PAGINATION#` substitution string, you would enter the following:

```
<table>#TOP_PAGINATION#
```

You can also include the substitution string `#CSV_LINK#` to include support for exporting your report to comma-separated value (CSV) format, a format compatible with most spreadsheet programs.

You can also use the substitution string `#REPORT_ATTRIBUTES#`. This substitution string is set per report region on the report attributes page.

Column Headings

Use **Before Column Heading** to display text before report heading.

Use **Column Heading Template** to colorize each column header cell. The text of this attribute must include Help to indicate where the cell heading text should be colorized. For example:

```
<th >Item Help Text</th>
```

If you do not want any column headings, enter the following:

```
OMIT
```

If you do use this attribute, Application Express engine applies the default column heading template.

Use **After Column Heading** to display text after report headings.

Before Each Row

In **Before Each Row**, enter text to display before all columns in the report. Use this attribute to open a new HTML row. Before Each Row supports the following substitution strings:

- `#ROWNUM#`

Use this substitution string to specify the current row.

- `#COLCOUNT#`

Use this substitution string to specify the number of columns.

- `#HIGHLIGHT_ROW#`

Use this substitution string to specify the number of highlighted rows.

Column Templates

Column templates define the look of each column. You can define up to four column templates; each can be conditional. For example, you can have different background colors for even and odd rows, or highlight rows that meet a PL/SQL defined condition.

In each Column Template, you define the look of each column. Column Templates support the substitution strings described in [Table 12-14](#).

Table 12-14 Column Template Substitution Strings

| Substitution String | Description |
|---------------------|---|
| #ALIGNMENT# | Determines the column alignment. Specified by the user. |
| #COLCOUNT# | Defines the count of the number of columns. |
| #COLNUM# | Defines the current column number. |
| #COLUMN_HEADER# | Defines the column header. |
| #COLUMN_VALUE# | Replaced with the value of the column. |
| #ROWNUM# | Specifies the current row number. |

Consider the following example:

```
<td> #ALIGNMENT#>#COLUMN_VALUE#</td>
```

If you actually ran this report, these substitution strings would be replaced with values generated by the results of a SQL query.

By creating conditions, you can create a report that displays columns differently depending on whether the specified condition is met. To specify a column template be used conditionally, select a condition type from the Column Template Condition list. Valid values include:

- **Use Based on PL/SQL Expression.** Conditionally formats columns based on data in that row.
- **Use for Even Numbered Rows.** Conditionally formats even numbered rows.
- **Use for Odd Numbered Rows.** Conditionally formats odd numbered rows.

If you select **Use Based on PL/SQL Expression**, the next step is to enter a PL/SQL expression in the Column Template Expression field. For example, the following expression displays a value in bold if the value is greater than 2000:

```
#SAL# > 2000
```

You could also use the substitution string #ROWNUM#. For example:

```
#ROWNUM# > 2000
```

After Each Row

In **After Each Row**, enter HTML that displays after all columns in the report display. This attribute is often used to close an HTML table row. For example:

```
</tr>
```


After Rows

Use **After Rows** to specify text that should display after the last row. A common use of this attribute is to close the HTML table tag. For example:

```
</table>
```

The After Rows attribute supports the following substitution strings:

- #PAGINATION#
Replaced with a pagination attribute.
- #COLCOUNT#
Substituted at runtime with the number of columns defined in the report.
- #CSV_LINK#
Substituted at runtime with CSV download link if CSV export is enabled. Otherwise the string is not shown.
- #EXTERNAL_LINK#
Substituted at runtime with the download link for exports using external processing engine for (for example, PDF export). If this feature is not enabled, the string is not shown.

Row Highlighting

Background color for checked row controls the background color of a report row when the row selector is checked. **Background color for current row** controls the background color of a report row when the user moves the mouse over the row.

Pagination Subtemplate

The Pagination Subtemplate section contains attributes for editing the Pagination Template, Next Page Template, Previous Page Template, Next Set Template, and Previous Template. Pagination Subtemplates support the substitution strings #PAGINATION_NEXT#, #PAGINATION_NEXT_SET#, #PAGINATION_PREVIOUS# and #PAGINATION_PREVIOUS_SET#. [Table 12-17](#) describes these templates.

Table 12-15 Pagination Subtemplate Attributes

| Pagination Subtemplate Attribute | Description |
|----------------------------------|--|
| Pagination Template | <p>Applies to the entire pagination subtemplate. For example:</p> <pre>#TEXT#</pre> <p>Use the substitution string #TEXT# to specify where you want the pagination subtemplate to display.</p> <p>Use the other Pagination Subtemplate attributes to modify individual items.</p> |

Table 12-15 (Cont.) Pagination Subtemplate Attributes

| Pagination Subtemplate Attribute | Description |
|----------------------------------|---|
| Next Page Template | Enter HTML to modify how the Next Page portion of the pagination subtemplate appears. For example: <code>next</code> |
| Previous Page Template | Enter HTML to modify how the Previous Page portion of the pagination subtemplate appears. For example: <code>previous</code> |
| Next Set Template | Enter HTML to modify how the Next Set portion of the pagination subtemplate appears. For example: <code>next set</code> |
| Previous Set Template | Enter HTML to modify how the Previous Set portion of the pagination subtemplate appears. For example: <code>previous set</code> |

Comments

Use this attribute to record comments about this component.

Report Column Template Attributes for Named Column Templates

This section describes specific sections of the Edit Report Template page for Named Column Templates. You can access the sections of the page by either scrolling down the page, or by clicking a navigation button at the top of the page. When you select a button at the top of the page, the selected section appears and all other sections are temporarily hidden. To view all sections of the page, click **Show All**.

- [Name](#)
- [Subscription](#)
- [Row Templates](#)
- [Column Headings](#)
- [Before first and after last row text](#)
- [Pagination](#)
- [Comments](#)

Name

Template Name identifies the name of the template. **Template Type** indicates the type of template. Named Column templates reference column names in the template. Generic Column Templates reference the #COLUMN_VALUE# substitution string in the template.

Theme indicates the theme to which the template is a member. Use the **Translatable** check box to indicate the template contains text strings that require translation.

Template Class identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Subscription

Use Subscription to apply an existing template to the current application. When you select an existing template, you become a subscriber to that template.

To load a new copy of a master template, click **Refresh**.

Row Templates

Row templates define the look of each column. You can define up to four row templates, each of which can be conditional.

In each Row Template, you define the look of each row. Row Templates support the substitution strings described in [Table 12-16](#).

Table 12-16 Row Template Substitution Strings

| Substitution String | Description |
|---------------------|--|
| #ALIGNMENT# | Determines the row alignment. Specified by the user. |
| #COLCOUNT# | Defines the count of the number of columns. |
| #COLNUM# | Defines the current column number. |
| #COLUMN_HEADER# | Defines the column header. |
| #COLUMN_VALUE# | Replaced with the value of the column. |
| #ROWNUM# | Specifies the current row number. |

By creating conditions, you can create a report that displays rows differently depending on whether the specified condition is met. To specify a row template be used conditionally, select a condition type from the Column Template Condition list. Valid values include:

- **Use Based on PL/SQL Expression.** Conditionally formats columns based on data in that row.
- **Use for Even Numbered Rows.** Conditionally formats even numbered rows.
- **Use for Odd Numbered Rows.** Conditionally formats odd numbered rows.

If you select **Use Based on PL/SQL Expression**, the next step is to enter a PL/SQL expression in the Column Template Expression field. For example, the following expression displays a value in bold if the value is greater than 2000:

```
#SAL# > 2000
```

You could also use the substitution string #ROWNUM#. For example:

```
#ROWNUM# > 2000
```

Column Headings

Use this template to add color to each column header cell. The text of this attribute must include help to indicate where the cell heading text should be colorized. If you do not enter a Column Heading Template, then a default column header template is applied. If you do not want any column headings, then enter OMIT. For example:

```
<th #ALIGNMENT#>#COLUMN_HEADER#</th>
```

Before first and after last row text

In **Before Rows**, enter HTML that displays once at the beginning of a report template. Opening an HTML table is a common use of this attribute, as shown in the following example:

```
<table>
```

You can identify column headers using the syntax #1#, #2#, #3#. For example:

```
<th>#1#</th><th>#2#</th><th>#3#</th>
```

You can include pagination above a report by including the substitution string #TOP_PAGINATION#. This substitution string generates HTML that starts with an opening <tr> tag and ends with a closing </tr> tag. For example, to include an open table tag and #TOP_PAGINATION# substitution string, you would enter the following:

```
<table>#TOP_PAGINATION#
```

You can also include the substitution string #CSV_LINK# to include support for exporting your report to CSV format, a format compatible with most spreadsheet programs.

Use **After Rows** to specify text that should display after the last row. A common use of this attribute is to close the HTML table tag. For example:

```
</table>
```

The After Rows attribute supports the following substitution strings:

- #PAGINATION#
Replaced with a pagination attribute.
- #COLCOUNT#
Substituted at runtime with the number of columns defined in the report.

Pagination

The Pagination section contains attributes for editing the Pagination Template, Next Page Template, Previous Page Template, Next Set Template, and Previous Template. Pagination Subtemplates support the substitution strings #PAGINATION_NEXT#, #PAGINATION_NEXT_SET#, #PAGINATION_PREVIOUS# and #PAGINATION_PREVIOUS_SET#. [Table 12-17](#) describes these templates.

Table 12-17 Pagination Subtemplate Attributes

| Pagination Subtemplate Attribute | Description |
|----------------------------------|--|
| Pagination Template | <p>Applies to the entire pagination subtemplate. For example:</p> <pre>#TEXT#</pre> <p>Use the substitution string #TEXT# to specify where you want the pagination subtemplate to display.</p> <p>Use the other Pagination Subtemplate attributes to modify individual items.</p> |
| Next Page Template | <p>Enter HTML to modify how the Next Page portion of the pagination subtemplate appears. For example:</p> <pre>next</pre> |
| Previous Page Template | <p>Enter HTML to modify how the Previous Page portion of the pagination subtemplate appears. For example:</p> <pre>previous</pre> |
| Next Set Template | <p>Enter HTML to modify how the Next Set portion of the pagination subtemplate appears. For example:</p> <pre>next set</pre> |
| Previous Set Template | <p>Enter HTML to modify how the Previous Set portion of the pagination subtemplate appears. For example:</p> <pre>previous set</pre> |

Comments

Use this attribute to record comments about this component.

About Using JavaScript in Column Templates

You can conditionally display HTML depending upon values in the database using JavaScript. The following example displays an HTML row only if the `GROUP_DESC` query column is not null:

```
<script language="javascript">
IF ( "#GROUP_DESC#" != "" )
document.writeln( "<TR>;
<TD BGCOLOR=#336699>;</TD>
</TR>
</TR>
<TD>#GROUP_DESC#</TD>
      </TR>" );
</TR>" );
```

 **See Also:**
["About Regions"](#)

Using Custom Cascading Style Sheets

Upload or reference a cascading style sheet.

A cascading style sheet (CSS) enables a developer to control the style of a web page without changing its structure. When used properly, 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 display.


Tip:

If your application uses *Universal Theme - 42*, you can also take advantage of theme styles and update the appearance of your application using Theme Roller. See ["Using Theme Styles and Theme Roller"](#) and ["Using Theme Roller."](#)

- [About Uploading Cascading Style Sheets](#)
- [Referencing an Uploaded Cascading Style Sheet in the Page Template](#)

About Uploading Cascading Style Sheets

You can upload files (including CSS files) for use by a specific application or all applications within a workspace. To learn more, see ["Managing Static Application Files"](#) and ["Managing Static Application Files."](#)

 **See Also:**
["Referencing an Uploaded Cascading Style Sheet in the Page Template"](#)

Referencing an Uploaded Cascading Style Sheet in the Page Template

You can reference an uploaded cascading style sheet within the Header section of the page template. You use the Header section to enter the HTML that makes up the <HEAD> section of the HTML document.

To reference an uploaded cascading style sheet:

1. On the Workspace home page, click the **App Builder** icon.

2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Themes**.
The Themes page appears.
5. Select a theme.
6. On the Tasks list, click **View Templates**.
7. Select the name of the page template you want to edit.
8. Use a `<link>` tag within the Header section to reference the appropriate style sheet.

To reference an uploaded file that is associated with a specific application, you would use the substitution string `#APP_IMAGES#`. For example:

```
<html>
<head>
  <title>#TITLE#</title>
  #HEAD#
  <link rel="stylesheet" href="#APP_IMAGES#sample2.css" type="text/css">
</head>
...
```

To reference an uploaded file that is associated with a specific workspace, you would use the substitution string `#WORKSPACE_IMAGES#`. For example:

```
<html>
<head>
  <title>#TITLE#</title>
  #HEAD#
  <link rel="stylesheet" href="#WORKSPACE_IMAGES#sample3.css" type="text/css">
</head>
...
```

See Also:

- ["About Uploading Cascading Style Sheets"](#)
- ["Creating a New Template"](#)
- ["Managing Templates"](#)
- ["Page Templates"](#)
- ["APP_IMAGES"](#)
- ["WORKSPACE_IMAGES"](#)

13

Controlling Page Layout

Developers can customize their database application's page layout by using a global page, customizing regions, and editing item attributes.

This section also describes how to manage images and static files as well as explains how to render HTML using custom PL/SQL.

- [About Page Layout in Oracle Application Express](#)
The Application Express engine renders pages by combining templates with application components defined by the developer and data in the database.
- [Optimizing a Page for Printing](#)
You can optimize a page for printing by creating a specific Printer Friendly template and selecting that template on the Create/Edit Theme page.
- [Creating a Global Page to Display Components on Every Page](#)
A global page functions as a master page.
- [Managing Regions](#)
Developers create regions to function as containers for content.
- [About Incorporating Content from Other Web Sites](#)
To incorporate content from other servers, you can create a region based on a URL to display content.
- [About Managing Images](#)
To reference an image within an application, you must upload it to the static file repository. During the upload process, you specify whether the file is available to all applications or just a specific application.
- [Rendering HTML Using Custom PL/SQL](#)
If you must generate specific HTML content not handled by Oracle Application Express forms, reports, and charts, you can use the PL/SQL region type.

See Also:

- ["Adding Navigation"](#)
- ["Managing the Application User Interface"](#)
- ["Using Themes and Theme Styles"](#)

About Page Layout in Oracle Application Express

The Application Express engine renders pages by combining templates with application components defined by the developer and data in the database.

The overall framework (or structure of a page) is determined by the page template. For example, the page template controls if a page uses tabs and a navigation bar. It can

also define if a page includes a bar on the left side that serves as a placeholder for navigation or secondary content. Finally, a page template can include definitions of region positions, which enable precise control over placement of regions using HTML tables or style sheet definitions. The page template itself is composed of HTML combined with substitution strings, which are substituted with the appropriate components at runtime.

As a developer, you add content to a page by creating a region. A **region** is an area of a page that serves as a container for content. Each region contains a different type of content such as HTML, a report, a form, a chart, a list, a breadcrumb, PL/SQL, a tree, a URL, or a calendar. You position a region either relative to other regions (that is, based on its sequence number and column), or by using a region position defined in the page template. The style of the region is also controlled by the region template. Like the page template, the region template defines the structure of the area that the region takes up on a page. It defines if the region title is displayed and where it is displayed relative to the main content or the body. A region can also define absolute positions for buttons.

 **See Also:**
["Using Themes and Theme Styles"](#)

Optimizing a Page for Printing


You can optimize a page for printing by creating a specific Printer Friendly template and selecting that template on the Create/Edit Theme page.

Generally, a Printer Friendly template optimizes a page for printing. For example, a Printer Friendly template might:

- Not display tabs or navigation bars
- Display items as text instead of as form elements

If the theme you select does not include a printer friendly template, you can create a Printer Friendly template by creating a new page template.

- [Selecting a Printer Friendly Template for an Application](#)
- [Using f?p Syntax to Toggle to Printer Friendly Mode](#)

 **See Also:**
["Editing Themes"](#) and ["Creating a Theme"](#)

Selecting a Printer Friendly Template for an Application

To select a Printer Friendly template:

1. Navigate to the Themes page:

- a. On the Workspace home page, click the **App Builder** icon.
- b. Select an application.
- c. Click **Shared Components**.
- d. Under User Interface, select **Themes**.


The Themes page appears.

2. Select a theme.
3. Scroll down to **Component Defaults**.
4. Make a selection from the **Printer Friendly Page** list.
5. Confirm your changes and click **Apply Changes**.

 **See Also:**
["Editing Themes"](#)

Using f?p Syntax to Toggle to Printer Friendly Mode

Once you create a Printer Friendly template and select it, you can use f?p syntax to toggle to Printer Friendly mode. Including the ninth f?p syntax argument (`PrinterFriendly`) renders the page in printer friendly mode (optimizing printed output). For example, you could include this argument when coding a link or creating a navigation bar icon.

 **See Also:**
["About Using f?p Syntax to Link Pages"](#)

Creating a Global Page to Display Components on Every Page

A global page functions as a master page.

A global page (previously referred to as Page 0) functions as a master page. Developers can create a separate Global page for each user interface. The Application Express engine renders all components you add to a Global page on every page within your application. You can further control whether the Application Express engine renders a component or runs a computation, validation, or process by defining conditions.

- [Creating a Global Page](#)
- [Navigating to the Global Page](#)

 **See Also:**

- ["Managing Pages in a Database Application"](#)
- ["Understanding Conditional Rendering and Processing"](#)
- ["Available Conditions"](#)

Creating a Global Page

To create a Global page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
The Application home page appears.
3. Click the **Create Page** button.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page.
 - b. Select a page type - Select **Global Page**.

Note that the Global Page option only appears if the application does not have a Global page for the specified user interface.

5. For Page Number, enter an integer value that identifies a page within the application.
6. Click **Finish**.

 **See Also:**

["Managing the Application User Interface"](#)

Navigating to the Global Page

To navigate to the Global page:

1. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
2. Select an application.
The Application home page appears.
3. Select **Global Page** (or **Page Zero** if this is an older application).

Managing Regions

Developers create regions to function as containers for content.

- [About Regions](#)
- [Creating a Region](#)
- [Editing Regions](#)
- [Copying a Region](#)
- [Deleting Regions](#)

About Regions

Each page in an Oracle Application Express application contains one or more regions. A region is a area on a page that serves as a container for content. Each page can have any number of regions. You control the appearance of a region through a specific region template. The region template controls the look of the region, the size, determines whether there is a border or a background color, and what type of fonts display. A region template also determines the standard placement for any buttons placed in region positions.

You can use regions to group page controls (such as items or buttons). You can create simple regions that do not generate additional HTML, or create elaborate regions that frame content within HTML tables or images.

Regions display in sequence in the page template body or can be placed explicitly into region positions. The page structure can be defined using HTML tables or more commonly today using `DIV` tags.

See Also:

- ["Managing Regions "](#)
- ["Accessing Page Specific Utilities"](#)

Creating a Region

Tip:

You can also create a form or breadcrumb region from the Page Designer toolbar. On the Page Designer toolbar, click the **Create** menu and then selecting **Form Region** or **Breadcrumb Region**.

To create a region in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.

2. In the Gallery at the bottom of the central pane, locate the type of region you want to create. Pass the cursor over a control or component to view a tooltip that describes it.

 **Tip:**

In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Select **Add To** and then the location where you want to add the component.

3. Click and hold the mouse on the region to be created and drag it the desired location in the Layout tab.

When the mouse is over the appropriate location, the Layout tab displays as a darkened yellow tile. Release the mouse to drop the component. You can only drop components into appropriate drop positions, as determined by the component type.

Based on the type of component you add, Page Designer indicates what actions are required next. The Messages tab displays a red or yellow badge indicating the number of messages you need to address.

4. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save** or **Save and Run Page**.

**See Also:**

["Messages Tab"](#)

Editing Regions

A region is an area of a page that serves as a container for content. You can alter the appearance of a page by editing region attributes.

- [Editing Region Attributes](#)
- [Controlling Region Positioning](#)
- [Specifying a Region Header and Footer](#)
- [Enabling Users to Customize a Page](#)
- [Utilizing Region Caching in Page Designer](#)
- [Specifying a Static Region ID](#)

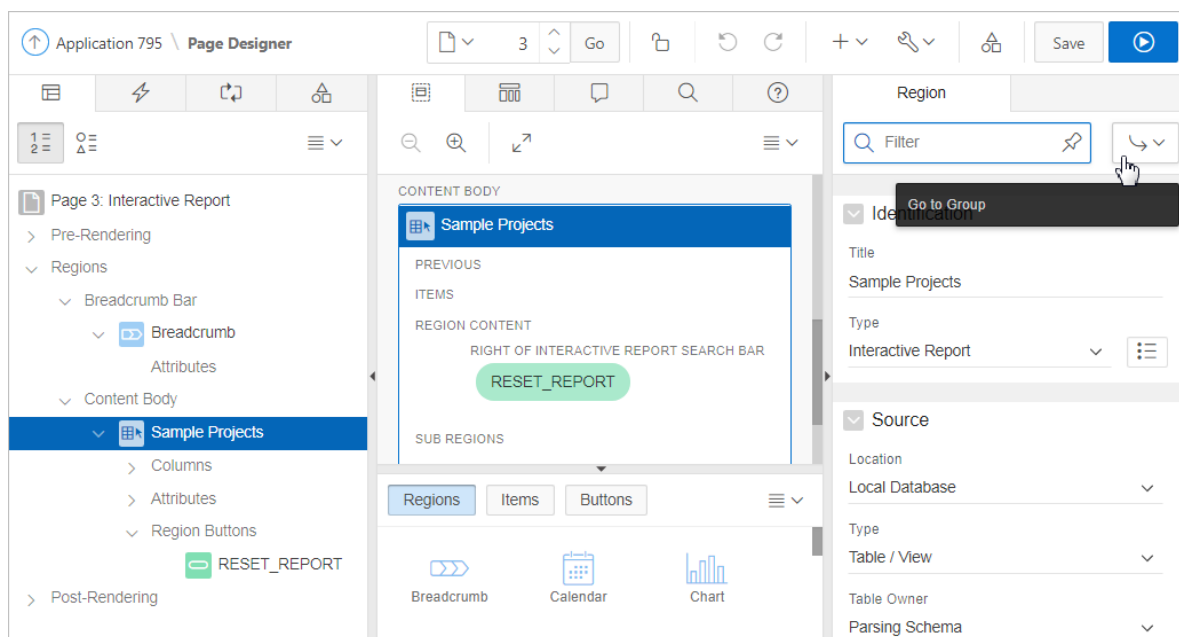
- [Adding a Region Image](#)
- [Creating a Region Display Selector](#)

Editing Region Attributes

To edit region attributes:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
 Page Designer appears.
2. In the Rendering tab, select the region.

The Property Editor displays the region attributes in the right pane.



3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes in the Property Editor.

To learn more about an attribute, see field-level Help.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**See Also:**

"Right Pane of Page Designer (Property Editor)"

Controlling Region Positioning

When you create a region, you can specify its position in the Property Editor by editing the Layout, Position attribute. The template positions defined within the current theme for the page's user interface determine the available selections.

Specifying a Region Header and Footer

In addition to the body content of a region, you can specify additional HTML to be placed above and below a region by editing the Header and Footer attributes in the Property Editor.

For all report regions, the substitution string `#TIMING#` shows the elapsed time in seconds used when rendering a region. You can use this substitution string for debugging purposes.

For classic report regions, the region footer supports the following substitution strings:

- `#ROWS_FETCHED#` shows the number of rows fetched by the Oracle Application Express reporting engine (the page size). You can use these substitution strings to display customized messages to the user. For example:

```
Fetches #ROWS_FETCHED# rows in #TIMING# seconds.
```

- `#TOTAL_ROWS#` displays the total number of rows that satisfy a SQL query used for a report.
- `#FIRST_ROW_FETCHED#` and `#LAST_ROW_FETCHED#` display the range of rows displayed. For example:

```
Row(s) #FIRST_ROW_FETCHED# through #LAST_ROW_FETCHED# of #ROWS_FETCHED# displayed
```

Enabling Users to Customize a Page

You can use the Customization attribute to control whether display attributes in a region can be customized by users. To use this feature, you must include the `#CUSTOMIZE#` substitution string in the Header, Body, or Footer section of the page template.

If at least one region supports end-user customization, a **Customize** link appears wherever you include the `#CUSTOMIZE#` substitution string in the page template. When users click this link, a window appears, enabling them to display or hide regions on the page.

To enable end user customization:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. Select the region in the Rendering tab or the Layout tab.

The Property Editor displays attributes for the page. Attributes are organized in groups.

3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Customization**. For Customizable, select one of the following:
 - **Not Customizable By End Users**
 - **Customizable and Not Shown By Default**
 - **Customizable and Shown By Default**
5. Click **Save**.



See Also:

["Creating Custom Themes"](#)

Utilizing Region Caching in Page Designer

Enabling region caching is an effective way to improve the performance of static regions such as regions containing lists that do not use conditions or regions containing static HTML.



Tip:

Region caching is not available for interactive report regions.

- [About Region Caching](#)
- [Enabling Region Caching in Page Designer](#)



See Also:

"Managing Page and Region Caching" in *Oracle Application Express Administration Guide*

About Region Caching

Enabling region caching is an effective way to improve the performance of static regions such as regions containing lists that do not use conditions or regions containing static HTML.

When you enable region caching, the Application Express engine renders a region from a cached (or stored) repository instead of rendering it dynamically. Keep in mind that the actual session identifiers are not cached. Instead, the Application Express engine caches a `&SESSION.` substitution string and the current session rendering the cached region is substituted on display. For example, if a region contains a link and the link includes a session, the exact session is not cached to ensure that the links works for all sessions.

The Application Express engine only renders a region from cache if it meets the defined condition. Additionally, regions can be cached specific to a user or cached independent of a user.

Enabling Region Caching in Page Designer

To enable region caching:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Select the region in the Rendering tab or the Layout tab.

The Property Editor displays attributes for the page. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Server Cache**. For **Caching**, select one of the following:
 - **Disabled** - Content is not cached and computed for each request.
 - **Enabled** - Content is cached and utilized by all users.
 - **Cached by User** - Content is cached specifically for each user.
 - **Cache by Session** - Content is cached specifically for each session.
5. Click **Save**.

Specifying a Static Region ID

Specifying a static region ID is useful when creating custom JavaScript or cascading style sheets. You can use the Static ID attribute on the Edit Region page to uniquely

identify a region. You can then reference the region using the #REGION_STATIC_ID# substitution string in a region templates, the header, the footer, or the body.

A static region ID is included by assigning it as an HTML element ID to the region container object (that is, the table or DIV). The ID of an HTML element must be unique for the entire page. For example, to keep the page HTML valid you cannot have a button and region with the same ID.

To specify a static region ID:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Select the region in the Rendering tab or the Layout tab.

The Property Editor displays attributes for the page. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Advanced**. Edit the **Static ID** attribute:
 - a. Find **Advanced** and then **Static ID**.
 - b. In Static ID, enter the static ID for this region. If defined, this is used as the ID for the region, using the substitution string #REGION_STATIC_ID#, which can be useful if developing custom JavaScript behavior for the region. If this is not defined, the region generates an internal ID.

 **Tip:**

The template must include the #REGION_STATIC_ID# substitution string, in order for the region to utilize the static ID entered.

5. Click **Save**.

Adding a Region Image

Use the Region Image and Image tag attributes to add an image to the upper left side of a region.

To add a region image:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Select a page.
Page Designer appears.
2. Select the region in the Rendering tab or the Layout tab.
The Property Editor displays Region attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Advanced**. Edit the following attributes:
 - **Region Image** - Enter a reference to an image that displays in the upper left of the region. There is no control over the table tag used to display this image.
 - **Image Tag Attributes** - Enter image tag attributes included in the image HTML. This can be used for attributes such as height, width or image ALT text (if the image is non-decorative).

If no ALT is defined in the image tag attributes, Application Express defaults to rendering the image as a decorative image, for example with an empty ALT tag.
5. Click **Save**.

Creating a Region Display Selector

Region Display Selector region enables you to include show and hide controls for each region on a page.

To create a Region Display Selector:

1. Create the page and regions to be included in the Region Page Selector.
2. For each region to be included in the selector, set the Region Display Selector attribute to **Yes**.
3. In the Gallery, right-click **Region Display Selector** region, select **Add To**, and select the appropriate location.

Tip:

You can also drag and drop from the Gallery. From the Gallery select the **Region Display Selector** region and drag it to the appropriate location in the Layout tab.

4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

5. Edit the region attributes in the Property Editor:
 - a. **Identification, Title** - Enter a region title. The region title only displays when it is defined in the region template.
 - b. **Advanced, Region Display Selector** - Select **Yes** if you want this region included in the Region Display Selector. This attribute is only utilized if a region type of Region Display Selector is defined and viewable on the page.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. Click **Save**.

Copying a Region

 **Note:**

You cannot copy a Tree region since this type of region encompasses more than one region.

To copy a region:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Right-click the region in Layout tab, select **Copy To**, and select a destination from the submenu.

 **Tip:**

To copy a region to another page, select the region in the Rendering tab, right-click, and select **Copy to Other Page**. Follow the on-screen instructions.

Deleting Regions

To delete regions:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Select a page.
Page Designer appears.
2. Select one or more regions in the Rendering tab.
3. Right-click and select **Delete**.
4. Click **Save**.

About Incorporating Content from Other Web Sites

To incorporate content from other servers, you can create a region based on a URL to display content.

Typically, pages in an application are based on data stored in an Oracle database. For example, suppose you wanted to reference the current Oracle stock price. You could create a region of type URL based on a URL. For example:

```
http://quote.yahoo.com/q?d=b&s=ORCL
```

You could then create an item called STOCK_SYMBOL and base your region on a stock price entered by the user. For example:

```
http://quote.yahoo.com/q?d=b&s=&STOCK_SYMBOL.
```

Sometimes (as is the case with the previous example) the HTML returned to the region is more than is needed. To restrict the HTML displayed, you can use the following region attributes:

- URL (discard until but not including this text)
- URL (discard after and including this text)

Tip:

The previous example may require that you set the Proxy Server application attribute. If you do not set the Proxy Server application attribute, you get an error message. Oracle Application Express uses the Oracle `utl_http.request_pieces` function to obtain the HTML generated from the given URL. See "[Editing Application Attributes](#)."

Developers can also use the Region attribute **Inclusion Mode** to control how Oracle Application Express retrieves and displays the content. Options include:

- **IFrame** - The browser displays the remote content in an HTML iframe. Oracle Application Express does not have to fetch this content into the database for display.
- **Inline (escape special characters)** - Causes the database to load the remote content at page rendering time. It then displays the fetched data inline, at the position of the region. Special HTML characters in the content are escaped, to prevent cross site scripting.
- **Inline (no escaping)** - This mode is similar to **Inline (escape special characters)** except the HTML characters are not escaped. Developers should only use this mode if the remote content is always safe.

 **See Also:**

["Editing Region Attributes "](#)

Working with SSL-Enabled URLs

If you call a SSL-enabled URL (for example, by invoking a Web service), or create a region of type URL that is SSL-enabled, you must create a wallet. A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL.

 **Tip:**

See "Configuring Wallet Information" in *Oracle Application Express Administration Guide*

About Managing Images

To reference an image within an application, you must upload it to the static file repository. During the upload process, you specify whether the file is available to all applications or just a specific application.

 **See Also:**

["Managing Static Application Files"](#) and ["Managing Static Workspace Files"](#)

Rendering HTML Using Custom PL/SQL

If you must generate specific HTML content not handled by Oracle Application Express forms, reports, and charts, you can use the PL/SQL region type.

To generate HTML in this type of region, you need to use the PL/SQL Web Toolkit. You can reference session state using bind variable syntax. Keep in mind that when you generate HTML in this way, you do not get the same consistency and control provided with templates.

To give you more control over HTML dynamically generated within a region, you can use PL/SQL. For example, to print the current date, you could create a region with the following source:

```
http.p(TO_CHAR(SYSDATE, 'Day Month DD, YYYY'));
```

This next example accesses tables:

```
DECLARE
    l_max_sal NUMBER;
BEGIN
    SELECT max(sal) INTO l_max_sal FROM emp;
```

```
    http.p('The maximum salary is: '||TO_CHAR(l_max_sal,'999,999.00'));  
END;
```



See Also:

Oracle Database Development Guide

Managing Database Application Controls

Developers can create and manage a variety of database application page controls, including page-level items, dynamic actions, buttons, and trees.

- [Managing Page-Level Items](#)
An item is part of an HTML form. Examples of page-level item include a check box, a date picker, plain text, a file browse field, a popup list of values, a select list, a text area, and so on.
- [Managing Dynamic Actions](#)
Dynamic actions enable developers to define complex client-side behavior declaratively without the need for JavaScript.
- [Managing Buttons](#)
You can use buttons to direct users to a specific page or URL, or to post or process information (for example, by creating Create, Cancel, Next, Previous, or Delete buttons). You can also configure buttons to display conditionally or warn users of unsaved changes.
- [Managing Trees](#)
Trees to display hierarchical information in a clear, easy-to-use format. You can create a tree control using a SQL query.
- [About Incorporating JavaScript into an Application](#)
Oracle Application Express includes multiple built-in interfaces especially designed for adding JavaScript.

See Also:

- ["Controlling Page Layout"](#)
- ["Adding Navigation"](#)
- ["Manually Refreshing Oracle Application Express Components"](#)

Managing Page-Level Items

An item is part of an HTML form. Examples of page-level item include a check box, a date picker, plain text, a file browse field, a popup list of values, a select list, a text area, and so on.

This section describes what a page-level item is, how to view existing page-level items, how to create new page-level items, and how to edit an item in Page Designer to change its appearance or behavior.

- [Understanding Page-Level Items](#)
An item is part of an HTML form such as a check box, date picker, display as text, file browse field, popup list of values, select list, or a text area.

- [Viewing Page-Level Items](#)
Learn how to view page-level attributes in Page Designer.
- [Creating Page-Level Items](#)
You create page-level items in Page Designer in either the Rendering tab or by adding an item from the Gallery.
- [Editing Page-Level Items](#)
Developers edit Item attributes in Page Designer to control how items display on a page.

Viewing Page-Level Items

Learn how to view page-level attributes in Page Designer.

Item attributes affect the display of items on a page. For example, these attributes can impact where a label displays, how large an item is, and if the item displays next to or below the previous item.

To view page-level attributes in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.
The Property Editor changes to display the attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes in the Property Editor.

Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Click **Save**.
6. Click **Save and Run** to view the page.

Creating Page-Level Items

You create page-level items in Page Designer in either the Rendering tab or by adding an item from the Gallery.

- [Creating a Page-Level Item from the Rendering Tab](#)
- [Adding a Page-Level Item from the Gallery](#)
- [Creating a Static List of Values](#)
- [Creating a Cascading List of Values](#)
- [Creating a Shuttle Item on the Form Page in Page Designer](#)

Creating a Page-Level Item from the Rendering Tab

To create a page-level item from the Rendering tab:

1. View the page to contain the item in Page Designer.
2. If necessary, create a region to contain the item.
3. In the Rendering tab, right-click the region to contain the item and select **Create Page Item**.

The Property Editor displays attributes for the item. Attributes are organized in groups.

4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Edit the appropriate attributes in the Property Editor.

Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

6. Click **Save**.

See Also:

["Viewing a Page in Page Designer"](#) and ["About Regions"](#)

Adding a Page-Level Item from the Gallery

To create a page-level item by adding it from the Gallery:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.

2. If necessary, create a region to contain the item.
3. In the central pane, click **Layout** and then the **Items** tab in the Gallery.
Passing the cursor over an item displays a tooltip that describes it.
4. Select an item and drag it to the appropriate location in the Layout tab.

 **Tip:**

You also right-click the item, click **Add To**, and select the appropriate location.

The Property Editor displays Page Item attributes.

The Messages tab displays a red or yellow badge to identify messages you need to address. Selecting a message displays the associated attribute in the Property Editor. You must address red error message before you can save.

5. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. Click **Save**.

 **See Also:**

["Viewing a Page in Page Designer"](#) and ["About Regions"](#)

Creating a Static List of Values

One way to create a static list of values is to edit an item's List of Values definition. Note that this type of list of values is not reusable. As a best practice, create a list of values as a shared component whenever possible.

To create a static list of values:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Create a new item from the Rendering tab or by adding it from the Gallery.

The Property Editor displays the Page Item attributes. Attributes are organized in groups.

3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Under **Identification**:
 - a. **Name** - Enter the name of this item. Item names must be valid Oracle identifiers. Oracle recommends that item names be no longer than 30 bytes so they can be used as bind variables in SQL Queries.
 - b. **Type** - Select **Select List**.
6. Under **List of Values**:
 - a. **Type** - Select **Static Values**.
 - b. **Static Values** - Click the **Display1, Display2** button. A dialog appears.
In the Static Values dialog:
 - **Display Value** - Enter the Display Value for each entry.
 - **Return Value** - Enter the Return Value for each entry. If you do not include a Return Value, the return value equals the Display Value
 - **Move Up** and **Move Down** - Click the **Move Up** and **Move Down** arrow buttons to change the order.
 - **Sort** - Select **Yes** to sort the list at alphabetically at runtime.

 **Tip:**

See the sections that follow to view examples.

- c. Click **OK**.
7. Edit the remaining List of Values attributes as appropriate. To learn more about an attribute, select the attribute and click the **Help** tab in the central pane.
8. Click **Save** to save your changes.
9. Click **Save and Run** to view the page.

Example 14-1 Example 1: Four Values Displayed in Alphabetical Order

This example shows the Static Values dialog with four values defined: `Lion`, `Dog`, `Cow`, and `Cat`. The return value of each entry is capitalized. **Sort at Runtime** is set to **Yes** so that the list displays alphabetical order.

Static Values
×

Values

| Display Value | Return Value | | |
|---------------|--------------|---|-----|
| Lion | LION | × | ↓ |
| Dog | DOG | × | ↑ ↓ |
| Cow | COW | × | ↑ ↓ |
| Cat | CAT | × | ↑ ↓ |
| | | × | ↑ |

Sort

Sort at Runtime

In a running application this select list would look similar to the following illustration.

Static - Four Values ▼

Cat

Cow

Dog

Lion

Example 14-2 Example 2: Ten Values Displayed in the Order Listed

This example shows the Static Values dialog with ten values defined: 10, 15, 20, 25, 50, 100, 200, 500, 1000, and 10000 which display in the order listed. The return value of each entry equals the display value.

| Display Value | Return Value | | | |
|---------------|--------------|---|---|---|
| 10 | | X | | ↓ |
| 15 | | X | ↑ | ↓ |
| 20 | | X | ↑ | ↓ |
| 25 | | X | ↑ | ↓ |
| 100 | | X | ↑ | ↓ |
| 200 | | X | ↑ | ↓ |
| 500 | | X | ↑ | ↓ |

Sort at Runtime: Yes **No**

Buttons: Cancel, OK

In a running application this select list would look similar to the following illustration.

Static - Ten Values

- 10
- 15
- 20
- 25
- 50
- 100
- 200
- 500
- 1000
- 10000

Example 14-3 Example 3: A List of Values with Having Both a Return and Display Value

This example shows the Static Values dialog with two values defined. The first value has a **Display Value** of `Yes` and a **Return Value** of `Y`. The second value has a **Display Value** of `No` and a **Return Value** of `N`. **Sort at Runtime** is set to **No** to make sure Yes always displays first.

Static Values
✕

Values

| Display Value | Return Value | | |
|---------------|--------------|---|---|
| Yes | Y | ✕ | ↓ |
| No | N | ✕ | ↑ |
| | | ✕ | ↑ |

Sort

Sort at Runtime Yes

In a running application this select list would look similar to the following illustration.

Static - Return/Display
▼

Yes

No

See Also:

["Creating Lists of Values at the Application-Level"](#) and ["About Item Naming Conventions"](#)

Creating a Cascading List of Values

By creating a cascading list of values (LOV), one item on a page determines the list of values for another item. App Builder includes cascading LOV support for the following item types: Select List, Shuttle, Checkbox, Radio Group, Popup LOV, and List Manager.

You create a cascading LOV by creating two items. To populate the second item, you specify a Cascading LOV Parent Item when running the Create Item Wizard or by editing the item's attributes.

The following example demonstrates how to create two related select lists where the value of the first list populates the second.

To create a cascading list of values:

1. Create the first item, P1_DEPTNO:
 - a. Create a select list item.
 - b. In the Property Editor, edit the attributes:
 - Identification: Name - Enter `P1_DEPTNO`
 - Identification: Type - Select **Select List**.
 - Label: Label - Enter `Department`
 - c. In the Property Editor, edit the List of Values attributes:
 - List of Values: Type - Select **SQL Query**.
 - List of Values: SQL Query, enter:

```
SELECT dname as d,
       deptno as r
FROM dept
ORDER BY dname
```
 - List of Values: Display Null Value - Select **Yes**.
 - List of Values: Null Display Value - Enter:
- `Select Department -`
 - Accept the remaining defaults.
 - Click **Save**.
2. Create the second item, P1_EMPNO:
 - a. Create a select list item.
 - b. In the Property Editor, edit the attributes:
 - Identification: Name - Enter `P1_EMPNO`
 - Identification: Type - Select **Select List**.
 - Label: Label - Enter `Employee`
 - c. In the Property Editor, edit the List of Values attributes:
 - List of Values: Type - Select **SQL Query**.
 - List of Values: SQL Query, enter:

```
SELECT ename as d,
       empno as r
FROM emp
WHERE deptno = :P1_DEPTNO
```
 - List of Values: Display Null Value - Select **Yes**.
 - List of Values: Null Display Value - Enter:
- `Select Employee -`
 - Accept the remaining defaults.
 - Click **Save**.
3. Click **Save and Run** to view the page.

Two select lists appear. Making a selection in the Department select list, determines which individuals display in the Employees select list.

Creating a Shuttle Item on the Form Page in Page Designer

To create a shuttle on the form page:

This section demonstrates how to create shuttle item type. First, you first create a report and form on the `DEPT` table that shows which employees are assigned to a given department. Second, you create a shuttle item that lists employees alphabetically to make it easier to assign employees to a department.

To create a shuttle item on a form:

1. View the form page in Page Designer.
2. In Page Rendering, right-click region containing the form and select **Create Page Item**.
3. In the Property Editor, edit the following attributes:
 - a. Identification: Name - Enter the name of this item (for example, `PX_EMP_LIST`).
Where `PX` in the item name (for example `P2_EMP_LIST`) indicates the page on which the item resides.
 - b. Identification: Type - Select **Shuttle**.
4. In the Property Editor, edit the List of Values attributes:
 - a. List of Values: Type - Select **SQL Query**.
 - b. List of Values: SQL Query, enter:

```
SELECT ename, empno FROM emp ORDER BY 1
```
5. In the Property Editor, edit the Source attributes
 - a. Source: Type - Select **SQL Query (return colon separated value)**.
 - b. Source: Item Source Value - Enter the SQL Query that returns one or more rows to be used as the source for this item value. If the result contains multiple rows then the value from each row is formed into a single colon delimited value. For example:

```
SELECT empno FROM emp WHERE deptno = :P8_DEPTNO ORDER BY ename
```
6. Click **Save**.
7. Click **Save and Run** to view the page.

Editing Page-Level Items

Developers edit Item attributes in Page Designer to control how items display on a page.

For example, item attributes can determine where a label displays, how large an item is, and if the item displays next to or below the previous item. Item attributes also control item behavior such as the item's default value, whether the item displays conditionally, or if the item is read-only.

- [Editing Page Item Attributes in the Property Editor](#)
- [Defining Default Values for Page Items](#)
- [Configuring Page Item Security](#)

- [Creating a Quick Pick Selection](#)
- [Displaying Conditional Page Items](#)
- [Displaying Read Only Page Items](#)
- [Applying a Format Mask to an Item](#)
- [Configuring Item Attributes to Warn Users of Unsaved Changes](#)
- [Viewing Item Utilities](#)

Editing Page Item Attributes in the Property Editor

To edit item attributes in the Property Editor:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

The Property Editor displays attributes for the item. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes.

Edited attributes display a blue marker to the left of the attribute name until the page is saved.



Note:

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#)."

5. Click **Save**.

Defining Default Values for Page Items

You define default values for an item using the Default attribute. The default value is used when the item's value is not derived from session state and when the source value is NULL.

To define a default value for an item:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.

- b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.
The Property Editor displays the item attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. In the Property Editor, find the Default group.
5. Under Default, select a Type. Options include:
 - **Static Value** - In **Static Value**, specify a default value for this item. The default value is used when the item's value is not derived from session state and when the source value is null.
 - **PL/SQL Expression** - In **PL/SQL Expression**, enter the PL/SQL expression that produces the value for this item.
 - **PL/SQL Function Body** - In **PL/SQL Function Body**, enter the PL/SQL function body which returns the value for this item.
6. Click **Save**.

Configuring Page Item Security

To configure page item security:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.
The Property Editor displays the item attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Security**. Configure the following Security attributes.

Table 14-1 Property Editor - Page Item, Security

| Attribute | Description |
|--|--|
| Authorization Scheme | Select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed. See Also: " Providing Security Through Authorization " |
| Session State Protection | Select a Session State Protection level. Options include: <ul style="list-style-type: none"> • Unrestricted - The item can be set by passing the item in a URL or in a form. No checksum is required in the URL. • Checksum Required - User Level - The item can be set by passing the item in a URL that includes a checksum specific to the workspace, application and user. • Checksum Required - Session Level - The item can be set by passing the item in a URL that includes a checksum specific to the session. • Restricted - May not be set from browser - The item cannot be altered using the URL or POSTDATA. Select this option to restrict what can set the item value to internal processes, computations, and so on. This attribute only applies to items that are not used as data entry items and is always observed, even if Session State Protection is disabled. Use this attribute for page or application items that have the following Display As types <ul style="list-style-type: none"> – Display Only (Save State=No) – Text Field (Disabled, does not save state) See Also: " Preventing URL Tampering " |
| Store value encrypted in session state | Specify whether to encrypt this item when stored in session state. If the contents of an item contain sensitive data, then you should encrypt the value when it is stored in the Application Express session state management tables. Otherwise, anyone with rights to read the Application Express meta data tables could potentially write a query to extract this sensitive data. Values up to 4000 bytes in length can be encrypted. Attempts to encrypt values longer than 4000 bytes produce an error message. |
| Escape Special Characters | This attribute only displays with some item types. Select Yes to prevent Cross-Site Scripting (XSS) attacks. Select No if you want that HTML tags which are stored in the page item or in the entries of a list of value are actually rendered. See Also: " Understanding Cross-Site Scripting Protection " |

Table 14-1 (Cont.) Property Editor - Page Item, Security

| Attribute | Description |
|-----------------------|--|
| Restricted Characters | <p>Select how to restrict the characters that can be saved in session state. Application Express displays an error message if a user tries to save data that does not conform to the selected character restriction. Restricting the characters hardens application security and is a possible counter measure to cross-site scripting (XSS) attacks.</p> <p>Available options include:</p> <ul style="list-style-type: none"> • All characters can be saved. No restriction applies. • Whitelist for a-Z, 0-9 and space Only allow characters a-z, A-Z, 0-9, and space. • Blacklist HTML command characters (<>"). Do not allow reserved HTML characters • Blacklist &<>"';* =% and --: Do not allow &, <, >, ", /, ;, ", *, , =, % and "--" (PL/SQL comment). • Blacklist &<>"';* =% or -- and new line Do not allow &, <, >, ", /, ;, ", *, , =, %, "--", and new line characters <p>See Also: "Understanding Cross-Site Scripting Protection"</p> |

Creating a Quick Pick Selection

Quick picks enable users to select predefined values with just one click, rather than typing in text or selecting from a list of available options. If the item type supports quick pick selection, use the Quick Picks attributes to define up to ten selections that display under a give item. Clicking on a quick pick sets the value of item. The following example shows quick picks for 5%, 10%, 15%, and 20%.

To create a quick pick:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.
The Property Editor displays the item attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Under **Quick Picks**:

- a. **Show Quick Picks** - Select **Yes**.
 - b. **Link Attributes** - Enter HTML attributes that are associated with the display of each Quick Pick link
 - c. **Label** - Enter the label for the each Quick Pick. This label is the text the end user of the application sees.
 - d. **Value** - Enter the value associated with each Quick Pick. This value replaces the existing item value when the user clicks the quick pick.
5. Click **Save**.

Displaying Conditional Page Items

To display a conditional item:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.
The Property Editor displays the item attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Server-side Condition**. Select a condition **Type** and enter appropriate information in the field provided.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save**.

Displaying Read Only Page Items

To display a read-only item:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In either the Rendering tab or the Layout tab, select the item to edit.

The Property Editor displays the item attributes. Attributes are organized in groups.

3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Under **Read Only**, select a **Type**, and enter appropriate information in the field provided.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save**.

Applying a Format Mask to an Item

You can apply a format mask to an item by selecting the item in Page Designer and editing the **Format Mask** attribute. Developers commonly select a format mask to determine how numbers and dates display.

To apply a format mask to an item:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. In either the Rendering tab or the Layout tab, select the item to edit.

The Property Editor displays the item attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Under **Appearance**, **Format Mask** - Select a format mask.

 **Tip:**

Only apply number format masks to items that contain numbers and only apply date format masks to items that contain dates. Otherwise, an error is raised since the component contains a value that can not be converted using the specified format mask.

5. Click **Save**.

Configuring Item Attributes to Warn Users of Unsaved Changes

Developers can use the **Warn on Unsaved Changes** item attribute to specify if the page item should be included in the unsaved changes check. This check warns the user when they try to navigate away from a page and when the page contains unsaved changes. Items not based on a database column (for example, a search item) may not need to be checked. For those items, set Warn on Unsaved Changes to Do Not Check.

To configure the Warn on Unsaved Changes attribute:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

The Property Editor displays the item attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Advanced**.
5. For **Warn on Unsaved Changes**, select one of the following:
 - **Page Default** - Check for unsaved changes when the button is clicked if **Warn on Unsaved Changes** is enabled at the page-level.
 - **Do Not Check** - Changes to the item are ignored by the unsaved changes check. Select this option for items not based on a database column (for example, a search item).
6. Click **Save**.



Tip:

The **Warn on Unsaved Changes** attribute is implemented using the `apex.page.warnOnUnsavedChanges` API.



See Also:

["Configuring Page Attributes to Warn Users of Unsaved Changes"](#) and [Oracle Application Express API Reference](#)

Viewing Item Utilities

The Utilities page includes Grid Edit pages and reports that enable you to edit items on multiple pages within a selected application.

To access the Utilities page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities region, click **Item Utilities**.
5. Select one of the following reports:
 - **Grid Edit of all Item Labels**
 - **Grid Edit of all Item Help Text**
 - **Item Help Subscriptions**
 - **All Page Items**
 - **Password Items**
 - **Conditional Items**

Understanding Page-Level Items

An item is part of an HTML form such as a check box, date picker, display as text, file browse field, popup list of values, select list, or a text area.

When defining an item, developers must follow defined naming conventions and follow specific rules when referencing item values stored in session state.

- [About the Differences Between Page Items and Application Items](#)
Page items are placed on a page and have associated user interface properties and Application items are not associated with a page.
- [About Item Naming Conventions](#)
When creating an item name, developers must follow very specific item naming conventions.

- [Referencing Item Values](#)
You can reference item values stored in session state in regions, computations, processes, validation, and branches.
- [About Referencing Items Using JavaScript](#)
When you reference an item, the best approach is to reference by ID.
- [Working with Multiple Select List Item](#)
Learn how to handle values returned from a multiple select list item.

About the Differences Between Page Items and Application Items

Page items are placed on a page and have associated user interface properties and Application items are not associated with a page.

There are two types of items: page items and application items. **Page items** are placed on a page and have associated user interface properties, such as Display Only, Label and Label Template. Examples of page-level items include a check box, date picker, display as text, file browse field, popup list of values, select list, or a text area. In contrast **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.



See Also:

["Managing Application-Level Items"](#)

About Item Naming Conventions

When creating an item name, developers must follow very specific item naming conventions.

When specifying an item name, remember the following rules. Item names must:

- Be unique within an application.
- Not include quotation marks.
- Begin with a letter or a number, and subsequent characters can be letters, numbers, or underscore characters.
- Be case-insensitive.
- Should not exceed 30 characters. Items longer than 30 characters cannot be referenced using bind variable syntax. See "[Referencing Session State Using Bind Variable Syntax](#)."
- Cannot contain letters outside the base ASCII character set.

As a best practice Oracle recommends including the page number when naming items. By default, wizards prefix page item names with `P<page no>_<item name>` (for example, `P1_NAME`).

Referencing Item Values

You can reference item values stored in session state in regions, computations, processes, validation, and branches.

[Table 14-2](#) describes the supported syntax for referencing item values.

Table 14-2 Syntax for Referencing Item Values

| Type | Syntax | Description |
|---------------------|-----------------------|--|
| SQL | :MY_ITEM | Standard bind variable syntax for items whose names are no longer than 30 bytes. Use this syntax for references within a SQL query and within PL/SQL code. |
| PL/SQL | V('MY_ITEM') | PL/SQL syntax referencing the item value using the V function. Use this syntax in PL/SQL code of packages or stored procedures and functions. Avoid this syntax in SQL statements. It may result in performance problems. |
| PL/SQL | NV('MY_NUMERIC_ITEM') | Standard PL/SQL syntax referencing the numeric item value using the NV function. Use this syntax in PL/SQL code of packages or stored procedures and functions. Avoid this syntax in SQL statements. It may result in performance problems. |
| Static Text (exact) | &MY_ITEM. | Static text. Exact Substitution. Note: Exact substitution syntax should be avoided in SQL or PL/SQL code because it can result in SQL Injection vulnerabilities. |

You can set the value of an item in your application using any of the following methods:

- For page-level items, use the Source Attribute to set the item value.

From the page, select the item name to view the Edit Page Item page. Scroll down to Source and edit the appropriate fields.

You can also set the value of an item in any region based on PL/SQL or a process using the following syntax:

```
BEGIN
  :MY_ITEM := 'new value';
END;
```

- Pass the value on a URL reference using f?p syntax. For example:

```
f?p=100:101:10636547268728380919::NO::MY_ITEM:ABC
```

- Set the value using a computation. Computations are designed to set item values. For example:

```
TO_CHAR(SYSDATE, 'Day DD Month, YYYY');
```

- Use the PL/SQL API to set an item value within a PL/SQL context. For example:

```
APEX_UTIL.SET_SESSION_STATE('MY_ITEM', SYSDATE);
F
```

 **See Also:**

- ["Clearing Session State"](#)
- ["Understanding Cross-Site Scripting Protection"](#)
- ["Managing Session State Values"](#)

About Referencing Items Using JavaScript

When you reference an item, the best approach is to reference by ID.

If you view the HTML source of an Oracle Application Express page in a web browser, you would notice that all items have an id attribute. This ID corresponds to the name of the item, not the item label. For example, if you create an item with the name `P1_FIRST_NAME` and a label of `First Name`, the ID is `P1_FIRST_NAME`.

You can get and set item attributes and values using the JavaScript functions `$('P1_FIRST_NAME')` and `$('P1_FIRST_NAME', 'Joe');`. Consider the following example:

```
function showFirstName(){
    alert('First Name is ' +$( 'P1_FIRST_NAME' ))
};
function setFirstName(pFirstName){
    $( 'P1_FIRST_NAME', pFirstName);
};
```

These functions can be called by other JavaScript functions or with the Execute JavaScript code dynamic action.

 **See Also:**

["APEX_JAVASCRIPT"](#) in *Oracle Application Express API Reference*

Working with Multiple Select List Item

Learn how to handle values returned from a multiple select list item.

- [About Handling Values Returned from a Multiple Select List Item](#)
- [Using `APEX_UTIL.STRING_TO_TABLE` to Convert Selected Values](#)

About Handling Values Returned from a Multiple Select List Item

A multiple select item renders as a multiple select list form element which can be either a Multiselect List or Shuttle item type. When submitted, selected values are returned in a single colon-delimited string. You can handle values in this format in three ways:

- Using the `INSTR` function
- Using the `APEX_UTIL.STRING_TO_TABLE` function

- Creating a shuttle

Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values

Suppose you had a report on the `EMP` and `DEPT` tables that is limited by the departments selected from a Department multiple select list. First, you create the multiple select item, `P1_DEPTNO`, using the following query:

```
SELECT dname, deptno
FROM dept
```

Second, you return only those employees within the selected departments as follows:

```
SELECT ename, job, sal, comm, dname
FROM emp e, dept d
WHERE d.deptno = e.deptno
AND instr(':'||:P1_DEPTNO||':'||e.deptno||:') > 0
```

Next, assume you want to programmatically step through the values selected in the multiple select item, `P1_DEPTNO`. To accomplish this task, convert the colon-delimited string into a PL/SQL array using the `APEX_UTIL.STRING_TO_TABLE` function. The following example demonstrates how to insert the selected departments into an audit table containing the date of the query.

```
DECLARE
    l_selected APEX_APPLICATION_GLOBAL.VC_ARR2;
BEGIN
    --
    -- Convert the colon separated string of values into
    -- a PL/SQL array

    l_selected := APEX_UTIL.STRING_TO_TABLE(:P1_DEPTNO);

    --
    -- Loop over array to insert department numbers and sysdate
    --

    FOR i IN 1..l_selected.count
    LOOP
        INSERT INTO report_audit_table (report_date, selected_department)
        VALUES (sysdate, l_selected(i));
    END LOOP;
END;
```



See Also:

"STRING_TO_TABLE Function" in *Oracle Application Express API Reference*

Managing Dynamic Actions

Dynamic actions enable developers to define complex client-side behavior declaratively without the need for JavaScript.

- [About Dynamic Actions](#)
Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. Using the Dynamic Action Create wizard, you specify an action that is performed when a defined set of conditions occur. You can also specify which elements are affected by the action, and when and how they are affected.
- [About Dynamic Action Events](#)
You can define dynamic actions to fire based on events that happen on the page. Oracle Application Express includes four different categories of events: Browser events, Framework events, Component events, and Custom events. This section describes all supported events, including the internal JavaScript event name in brackets.
- [Creating a Dynamic Action](#)
Creating a dynamic action involves specifying when the action happens (with optional conditions), what action or actions are performed, and what elements are affected by the action. To learn more about any Property Editor attribute, select the attribute and click the **Help** tab in the central pane.
- [Editing Dynamic Actions](#)
Once you create a dynamic action, you can modify attributes defined during the creation process, specify attributes not available during the process (such as specifying an Authorization Scheme) and add additional true actions.
- [Defining Dynamic Action Event Scope](#)
After creating the dynamic action, the scope of the action can be modified to trigger only once, for the lifetime of the current page, or until triggering elements are updated by a Partial Page Refresh (PPR).
- [Deleting a Dynamic Action](#)
Delete a dynamic action by selecting it in Page Designer and selecting Delete from the context menu.
- [About Calling JavaScript Using a Dynamic Action](#)
You can execute JavaScript code by creating a dynamic action.
- [Debugging Dynamic Actions](#)
Learn how to debug dynamic actions in Oracle Application Express.

**See Also:**

["Accessing Page Specific Utilities"](#)

About Dynamic Actions

Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. Using the Dynamic Action Create wizard, you specify an action that is performed when a defined set of conditions occur. You can also specify which elements are affected by the action, and when and how they are affected.

When working with dynamic actions, you should be mindful of the fact that the more dynamic actions you add to a page, the greater your overall page size. This is because the dynamic action framework emits additional code to the client for each dynamic

action defined, which then also must be downloaded and executed by the framework in the client.

The process of implementing a dynamic action involves the following steps:

1. Edit or create an interactive grid column, item, button, region, JavaScript Expression, or jQuery selector on a page. This component is referenced within the dynamic action in defining when it fires.
2. Create a dynamic action from the application page that invokes the action.
3. Run your application to test the dynamic action.



Tip:

See "[Debugging Dynamic Actions](#)" for information on how to debug problems.

Viewing Dynamic Action Examples

To view dynamic action examples, install the *Sample Dynamic Actions* sample application. As an alternative, go to the Oracle Learning Library at <http://www.oracle.com/oll/apex>. Enter search criteria in the field provided (for example, dynamic actions) and click **Search**.



See Also:

"[Installing a Productivity and Sample App](#)"

About Dynamic Action Events

You can define dynamic actions can to fire based on events that happen on the page. Oracle Application Express includes four different categories of events: Browser events, Framework events, Component events, and Custom events. This section describes all supported events, including the internal JavaScript event name in brackets.

Browser Events



Note:

The events displayed differ according to the page's current User Interface type. If you want to select an event that corresponds to a different type, then you have the option of selecting Show unsupported, which displays all events including those that do not correspond to the current type.

- Change (`change`) - Fires when a control loses the input focus and its value has been modified since gaining focus.

- Click (`click`) - Fires when the pointing device button is clicked over the triggering element.
- Double Click (`dblclick`) - Fires when the pointing device button is double clicked over the triggering element.
- Double Tap (`apexdoubletap`) - Fires when the pointer is doing a double tap/click.
- Get Focus (`focusing`) - Fires when the triggering element receives focus by either a pointing device or by tabbing into the element.
- Key Down (`keydown`) - Fires when a key on the keyboard is pressed. Use this event when you want to capture special keystrokes such as arrow keys, after a key has been pressed.
- Key Press (`keypress`) - Fires when a key on the keyboard is pressed resulting in text being entered. Use this event when you want to capture actual text entry.
- Key Release (`keyup`) - Fires when a key on the keyboard is released. Use this event when you want to capture special keystrokes such as arrow keys, after a key has been released.
- Lose Focus (`focusout`) - Fires when the triggering element loses focus either by the pointing device or by tabbing out of the element.
- Mouse Button Press (`mousedown`) - Fires when the pointing device button is pressed over the triggering element.
- Mouse Button Release (`mouseup`) - Fires when the pointing device button is released over the triggering element.
- Mouse Enter (`mouseenter`) - Fires once when the pointing device is moved into the triggering element.
- Mouse Leave (`mouseleave`) - Fires once when the pointing device is moved away from the triggering element.
- Mouse Move (`mousemove`) - Fires when the pointing device is moved while it is over the triggering element.
- Pan (`apexpan`) - Fires when the pointer is down, then moved in a horizontal direction.
- Page Load `ready` – Fires when the page loads.
- Page Unload (`unload`) - Fires when a page is unloaded.
- Press (`apexpress`) - Fires when the pointer is down for greater than 250ms.
- Resize (`resize`) - Fires when the browser window is resized.
- Resource Load (`load`) - When the triggering element is the window element (using a JavaScript Expression value of `window` in the `When` attributes), the event fires when the browser finishes loading all content within a document, including window, frames, objects and images. For other elements, this event can only be used for elements associated with a URL: images, scripts, frames, iframes.
- Scroll (`scroll`) - Fires when a scrollable triggering element is scrolled. This could be the browser window (using a JavaScript Expression value of `window` in the `When` attributes), scrollable frames or elements with the `overflow` CSS property set to `scroll` (or `auto` when the element's explicit height is less than the height of its contents).
- Select (`select`) - Fires when a user selects some text in a text field.

- Swipe (`apexswipe`) - Fires when the pointer is moving fast in a horizontal direction.
- Tap (`apextap`) - Fires when the pointer is doing a small tap click.

Framework Events

- After Refresh (`apexafterrefresh`) - Fires after the triggering element has been refreshed. The event is only valid for triggering elements that perform Partial Page Refresh and fire this event. The native components that support this are Interactive Reports, Classic Reports, Charts, List View and all item types with cascading LOV support. Plug-ins might support this event as well.
- Before Page Submit (`apexbeforepagesubmit`) - Fires before a page being submitted.
- Before Refresh (`apexbeforerefresh`) - Fires before the triggering element has been refreshed. The event is only valid for triggering elements that perform Partial Page Refresh and fire this event. The native components that support this are Interactive Reports, Classic Reports, Charts, List View and all item types with cascading LOV support. Plug-ins might support this event as well.
- Dialog Closed (`apexafterclosedialog`) - Fires when an Application Express dialog is closed. This event only fires when the dialog is closed using the 'Close Dialog' page process, or the 'Close Dialog' dynamic action.

Component Events

These events are available when there is a component (either an item, region, or dynamic action) available to your application that triggers a custom event. These events appear in the following format Event name [Component Name], for example the Change Order event triggered by the Shuttle native item type appears as Change Order [Shuttle]. Component events are either triggered from native components shipped with Oracle Application Express, or from plug-in components you have installed into your application.

- Events triggered by native components:
Change Order [Shuttle] (`shuttlechangeorder`) – Fires when the order of a value in the right hand select list is changed (either using Move Top, Move Up, Move Down, or Move Bottom). There are currently no other events triggered by native components in Oracle Application Express.
- Events triggered by plug-in components:
These will be available when added to your current application and will be in the format Event name [Component Name]. For help related to events raised by plug-ins, refer to Help text on the plug-in configuration page, by navigating to Shared Components, Plug-ins, *plug-in name*, Help Text, where the plug-in author may have included documentation.
- Custom Event:
By selecting Custom an additional field displays enabling you to define of a custom event. This is useful when the native or plug-in provided events are insufficient.

Creating a Dynamic Action

Creating a dynamic action involves specifying when the action happens (with optional conditions), what action or actions are performed, and what elements are affected by

the action. To learn more about any Property Editor attribute, select the attribute and click the **Help** tab in the central pane.

To create a dynamic action in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane.
3. Under Dynamic Actions, right-click **Events** and select **Create Dynamic Action**.

**Tip:**

You can also select an event type and then right-click to select **Create Dynamic Action**.

The Property Editor displays Dynamic Action attributes.

The Messages tab displays a red or yellow badge to identify messages you need to address. Selecting a message displays the associated attribute in the Property Editor. You must address red error message before you can save.

4. In the Property Editor, edit the following Dynamic Action attributes:

**Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- a. **Identification, Name** - Enter the name of the dynamic action.
- b. **Execution Options, Sequence** - Enter the sequence for this computation. The sequence determines the order of execution.
- c. **When, Event** - Specify the event that causes the dynamic action to fire.
- d. **When, Selection** - Select the type of page element or construct to be used to trigger the event.

**Note:**

Only available if the event selected supports definition of a page element. Selecting any of the following events hides this attribute: Page Load, Page Unload, Resize, Before Page Submit, Orientation Change. All other event types show this field.

- e. **Client-side Condition, Type** - Optionally select the type of condition you want to control the true and false action processing of the dynamic action. If no client-side condition is defined, only true actions will fire. If a client-side condition is defined, the true action will fire when the condition is met, and the false action will fire when it is not.

 **Note:**

Other properties will display conditionally based on the type, to declaratively allow you to define the condition. For example for the condition type `Item = Value`, you will see an `Item` and `Value` property, which will be checked in evaluating the condition.

Next, define the action that to be performed if the event evaluates to True or False.

5. To edit an existing action:
 - a. Expand the Dynamic Action tree and select an existing action (that is, select either **True** or **False**).
 - b. In Property Editor, edit the following **Action** attributes:
 - **Action** - Specify which action you want to perform.
 - **Affected Elements** - Select how to define the page components to be affected when this action is executed. Additional options display depending upon the type of element selected.
 - **Execution Options, Fire When Event Result Is** - Specify whether this action fires when the triggering element condition is met by selecting True (True Action), or when it is not met by selecting False (False Action). If no condition is specified, only True actions fire.
 - **Selection Type** - Select how to define the page components to be affected when this action executes.
 - **Fire on Initialization** - Specify if the action fires on initialization.
Initialization has a slightly different meaning depending on how the dynamic action is defined. For dynamic actions defined to fire on interactive grid columns, this specifies if the action fires when the interactive grid row is activated for editing. For all other dynamic actions, this specifies if the action fires when the page loads.
6. To add a new action:
 - a. Expand the Dynamic Action tree.
 - b. Right-click the dynamic action and select either **Create TRUE Action** or **Create FALSE Action**.
 - c. Edit the action in the Property Editor as described in the previous step.
7. Click **Save**.

**See Also:**

["About Dynamic Action Events"](#)

Editing Dynamic Actions

Once you create a dynamic action, you can modify attributes defined during the creation process, specify attributes not available during the process (such as specifying an Authorization Scheme) and add additional true actions.

To edit a dynamic action:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane. Dynamic actions are organized by events.
3. Expand the appropriate event and select the dynamic action.
Attributes for the dynamic action display in the Property Editor.
4. In Property Editor, edit the appropriate Dynamic Action attributes.

**Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To edit an existing action:
 - a. Expand the Dynamic Action tree to view the True or False nodes.
 - b. Under True or False, select the action.
 - c. Edit the action in the Property Editor.
6. To add a new action:
 - a. Expand the Dynamic Action tree.
 - b. Right-click the dynamic action and select either **Create TRUE Action** or **Create FALSE Action**.
 - c. Edit the action in the Property Editor.
7. Click **Save**.

Defining Dynamic Action Event Scope

After creating the dynamic action, the scope of the action can be modified to trigger only once, for the lifetime of the current page, or until triggering elements are updated by a Partial Page Refresh (PPR).

To specify scope:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane.
3. Under Dynamic Actions, select the dynamic action.

The Property Editor displays attributes for the dynamic action. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Under **Advanced**, select **Event Scope**. The scope of the event determines when the event is evaluated a second or successive time. Options include:
 - **Static** - Binds the event handler to the triggering element(s) for the lifetime of the current page, but is no longer bound if a triggering element is updated via Partial Page Refresh (PPR).
 - **Dynamic** - Binds the event handler to the triggering element(s) for the lifetime of the current page, irrespective of any triggering elements being recreated via Partial Page Refresh (PPR).
 - **Once** - Binds the event handler to the triggering element(s) only once. The dynamic action is not triggered again until after the page has been fully refreshed.
6. Click **Save**.

Deleting a Dynamic Action

Delete a dynamic action by selecting it in Page Designer and selecting Delete from the context menu.

To delete a dynamic action:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Select a page.
Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane.
3. Expand the appropriate event and locate the dynamic action to be deleted.
4. Right-click the dynamic action and select **Delete**.
5. Click **Save**.

About Calling JavaScript Using a Dynamic Action

You can execute JavaScript code by creating a dynamic action.

You can also execute JavaScript code by creating a dynamic action with the Action **Execute JavaScript Code** and **Set Value**. You can also use JavaScript code for the condition of a dynamic action by setting the Client-side Condition type to **JavaScript Expression**.



See Also:

["Managing Dynamic Actions"](#)

Debugging Dynamic Actions

Learn how to debug dynamic actions in Oracle Application Express.

- [About Debugging Dynamic Actions](#)
- [Debugging Dynamic Actions](#)

About Debugging Dynamic Actions

Debugging dynamic actions in Oracle Application Express is slightly different than other debugging, because much of the processing done with the dynamic action framework is done on the client, not on the server. To debug dynamic actions, Oracle Application Express outputs debug information to the browser's JavaScript console if your browser supports it (for example Firefox with Firebug installed shows the debug information in its Console pane). The debug information tells you when an action of a dynamic action fires, along with some additional information about the dynamic action, in the following format:

```
Dynamic Action Fired: [Dynamic Action name] ([Action name]) {JavaScript object containing all Dynamic Action information}
```

This format enables you to identify the dynamic action name, the action name which indicates which action is triggered, and the JavaScript object which contains a lot of information about the dynamic action, including the when element, the affected elements, the event object and any data that may be associated with the dynamic action.

Debugging Dynamic Actions

To debug a dynamic action:

1. Ensure the application containing the dynamic action has Debugging enabled. See "[Utilizing Debug Mode](#)."
2. Run the page containing the dynamic action.
3. Open the browser's JavaScript console.
4. From the Developer toolbar, click **Debug**.

The page refreshes. If you have any dynamic actions that are set to fire on page load, you will see the debug output in the browser console.

Since debug information is only output when running in Debug mode, leaving Debug mode switched on enables you to further test if dynamic actions are firing when you expect them to. For example if you have defined a dynamic action that fires when a certain item's value changes, change that item's value and the console shows the debug output if the dynamic action fires.

Managing Buttons

You can use buttons to direct users to a specific page or URL, or to post or process information (for example, by creating Create, Cancel, Next, Previous, or Delete buttons). You can also configure buttons to display conditionally or warn users of unsaved changes.

- [What Actions Can a Button Perform?](#)
Use buttons to submit a page or redirect to a different page.
- [Creating a Button](#)
Create buttons in Page Designer.
- [Editing a Button](#)
Edit buttons in Page Designer.
- [Displaying a Button Conditionally](#)
You can choose to have a button display conditionally by editing the Server-side Condition attribute in Page Designer.
- [Configuring Button Attributes to Warn Users of Unsaved Changes](#)
Use the Warn on Unsaved Changes attribute to warn users of unsaved changes when they attempt to navigate away from a page.
- [About Calling JavaScript from a Button](#)
Call JavaScript from a button to confirm a request. Oracle Application Express uses this technique for the delete operation of most objects.
- [About the Relationship Between Button Names and REQUEST](#)
The name you give a Submit button (that is, a button with an Action of Submit Page) determines the value of the built-in attribute `REQUEST` when the page is submitted.
- [About Branching with Buttons](#)
Learn about branching with buttons.

**See Also:**

"[Calling a Page from a Button URL](#)" and "[Accessing Page Specific Utilities](#)"

What Actions Can a Button Perform?

Use buttons to submit a page or redirect to a different page.

Buttons can perform different types of actions. A button can:

- Submit a page (for example to save changes to a form page).
- Redirect to either a different page or a custom URL.
- Do nothing (for example if the button's behavior is defined in a Dynamic Action).

Creating a Button

Create buttons in Page Designer.

To create a button in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. If necessary, create a region to contain the item.
3. In the Gallery, click the **Buttons** tab.

Passing the cursor over a button displays a tooltip that describes it.
4. Right-click the button, select **Add To**, and select the appropriate location.

**Tip:**

You also select the button with the mouse and drag it to the appropriate location in the Layout tab.

The Property Editor displays Button attributes. Attributes are organized in groups.

5. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
6. Edit the appropriate attributes in the Property Editor.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Click **Save**.

 **See Also:**

["About Regions"](#)

Editing a Button

Edit buttons in Page Designer.

To create a region button in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the button to edit.
The Property Editor displays the button attributes in the right pane.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the button attributes.
Edited attributes display a blue marker to the left of the attribute name until the page is saved.

 **Note:**

To learn more about an attribute, see field-level Help.

5. Click **Save**.

Displaying a Button Conditionally

You can choose to have a button display conditionally by editing the Server-side Condition attribute in Page Designer.

To have a button display conditionally:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the button to edit.
The Property Editor changes to display Button attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Find **Server-side Condition**. Select a Type and enter appropriate information in the field provided.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save**.

 **See Also:**

["Referencing Session State Using Bind Variable Syntax"](#)

Configuring Button Attributes to Warn Users of Unsaved Changes

Use the Warn on Unsaved Changes attribute to warn users of unsaved changes when they attempt to navigate away from a page.

To configure the Warn on Unsaved Changes attribute:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.

- b. Select an application.
 - c. Select a page.
- Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the button to edit.
The Property Editor changes to display button attributes. Attributes are organized in groups.
 3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
 4. Find **Behavior**.
 5. For **Warn on Unsaved Changes**, select one of the following:
 - **Page Default** - Check for unsaved changes when the button is clicked if **Warn on Unsaved Changes** is enabled at page level.
 - **Do Not Check** - The unsaved changes check will not be performed when the button is clicked. Use this setting for Cancel, Delete, and Apply Changes buttons.
 6. Click **Save**.

**Tip:**

The **Warn on Unsaved Changes** attribute is implemented using the `apex.page.warnOnUnsavedChanges` API.

**See Also:**

["Configuring Page Attributes to Warn Users of Unsaved Changes"](#) and *Oracle Application Express API Reference*

About Calling JavaScript from a Button

Call JavaScript from a button to confirm a request. Oracle Application Express uses this technique for the delete operation of most objects.

For example, when you delete a button, a JavaScript message appears asking you to confirm your request. Consider the following example:

```
function deleteConfirm(msg)
{
var confDel = msg;
if(confDel !=null)
confDel= confirm("Would you like to perform this delete action?");
```

```

else
    confDel= confirm(msg);

if (confDel== true)
    doSubmit('Delete');
}

```

This example creates a function to confirm a delete action and then calls that function from a button. Note that the function optionally submits the page and sets the value of the internal variable `:REQUEST` to `Delete`, thus performing the deletion using a process that conditionally executes based on the value of the request.

When you create the button, you must select **Redirect to URL**. Then, you would specify a URL target such as the following:

```
javascript:confirmDelete('Would you like to perform this delete action?');
```

Oracle recommends using dynamic actions as the preferred way of executing JavaScript code. Consider the following example:

1. Create a button with action of **Defined by Dynamic Action**.

Create a dynamic action and using the action type **Execute JavaScript Code** to execute the previous code, for example:

```

if (confirm("Would you like to perform this delete action?")) {
    apex.submit('Delete');
}

```

This example uses JavaScript, but you could also easily implement this example without having to use JavaScript. Instead, you can use the declarative actions **Confirm** and **Submit Page** which are also translatable.



See Also:

["Creating a Button"](#) and ["Managing Dynamic Actions."](#)

About the Relationship Between Button Names and REQUEST

The name you give a Submit button (that is, a button with an Action of Submit Page) determines the value of the built-in attribute `REQUEST` when the page is submitted.

You can reference the value of `REQUEST` from within PL/SQL using the bind variable `:REQUEST`. By using this bind variable, you can conditionally process, validate, or branch based on which button the user clicks. You can also create processes that execute when the user clicks a button. And you can use a more complex condition as demonstrated in the following examples:

```

If :REQUEST in ('EDIT','DELETE') then ...
If :REQUEST != 'DELETE' then ...

```

These examples assume the existence of buttons named `EDIT` and `DELETE`. You can also use this syntax in PL/SQL Expression conditions. Be aware, however, that the button name capitalization (case) is preserved. In other words, if you name a button `LOGIN`, then a request looking for the name `Login` fails. For example:

```
<input type="BUTTON" value="Finish" onclick="apex.submit('Finish');">
```

In this example *Finish* is the name of the `REQUEST` and this example is case-sensitive.

About Branching with Buttons

Learn about branching with buttons.

Each page can include any number of branches. A branch links to another page in your application or to a URL. The Application Express engine considers branching at different times during page processing. You can choose to branch before processing, before computation, before validation, and after processing. Like any other control in App Builder, branching can be conditional. For example, you can branch when a user clicks a button. When you create a branch, you associate it with a specific button. The branch is only be considered if a user clicks the button.



See Also:

["Controlling Navigation Using Branches"](#)

Managing Trees

Trees to display hierarchical information in a clear, easy-to-use format. You can create a tree control using a SQL query.

- [About Trees](#)
Tree controls in Oracle Application Express use APEX Tree.
- [Creating a Tree on New Page](#)
Create a tree on a new page by running the Create Page Wizard.
- [Creating a Tree in Page Designer](#)
Create a tree control by providing a SQL query that specifies a hierarchical relationship by identifying an ID and parent ID column in a table or view.
- [Editing Tree Attributes](#)
Developers can customize tree behavior by editing tree attributes.

About Trees

Tree controls in Oracle Application Express use APEX Tree.

App Builder includes a built-in wizard for generating a tree hierarchical navigation mechanism. Trees are implemented using a single hierarchical query that identifies the row to be used as the start of your query and the relationship between parent rows and child rows of the hierarchy. Trees use the APEX Tree implementation. This is a JavaScript-based, cross browser tree component that features optional keyboard navigation, and optional state saving.

When you implement a tree control, the SQL query specifies a hierarchical relationship by identifying an ID and parent ID column in a table or view. The tree query must utilize a `START WITH ... CONNECT BY` clause to generate the hierarchical query.

 **Tip:**

The creation of APEX Tree regions is only supported on pages using a Desktop user interface.

About Desupported jsTrees

Prior to release 5.2, Oracle Application Express supported the rendering of jsTree tree regions. Oracle Application Express no longer supports jsTree tree regions. Oracle Application Express now supports the generation of an APEX Tree tree region. APEX Tree is a JavaScript based, cross browser tree component. jsTree regions are automatically upgraded to APEX Tree regions when upgrading from a previous release or when importing to release 18.1.

Viewing Tree Examples

To view tree examples, install the sample application, *Sample Trees*.

 **See Also:**

["Managing Productivity and Sample Apps in a Full Development Environment"](#)

Creating a Tree on New Page

Create a tree on a new page by running the Create Page Wizard.

A tree is based on a query and returns data that can be represented in a hierarchy. When you create a tree using the Create Page Wizard, the wizard generates the hierarchical query for you based on the options you select.

To create a tree on a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Select a page type - Select **Tree**.
5. For Page Attributes, specify the following:
 - a. Page Number - Specify the page on which the tree should be created.
 - b. Page Name - If the tree will be created on a new page, enter the page name.
 - c. Page Mode - Select a page mode.

- d. Page Group - Select the name of the page group you would like to associate with this page. This option will only be visible when the application contains groups.
 - e. Region Template - Select a region template for the tree region.
 - f. Region Name - Enter a name for the region to contain the tree.
 - g. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use. If you select Breadcrumb, enter the following:
 - Entry Name - Enter a name for the breadcrumb entry.
 - Select Parent Entry - Select a parent entry.
 - h. Click **Next**.
6. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
7. For Table/View Owner and Name:
 - a. Table/View Owner - Select the owner of the table from which the tree will be based.
 - b. Table / View Name - Select the table or view which contains the columns to be included in the master page.
 - c. Click **Next**.
8. For Query, identify the column you want to use as the ID, the Parent ID, and text that should appear on the nodes:
 - a. ID - Select the column to use as the ID.
 - b. Parent ID - Select the column to use as the parent ID.
 - c. Node Text - Select the text to appear on the tree nodes.
 - d. Start With - Select the column to be used to specify the root of the hierarchical tree query.
 - e. Start Tree - Choose how to start your query. Options include:
 - Based on Existing Item - Select an existing application or page item.
 - Based on a SQL Query - Enter a SQL query that returns a single row or single column.
 - Based on a Static Value - Enter a static value.
 - Value is NULL.
 - f. Click **Next**.
9. For Where and Order by, specify the following:
 - a. Where Clause - Enter a `WHERE` clause. To learn more, expand **Current Query**.
 - b. Order Siblings By - Select the order siblings by column, such as `ENAME`. The default value is based on the Node Text column selected.
 - c. Click **Next**.

10. For Tree Attributes, specify the following:
 - a. Include Buttons - Select the buttons to include.
 - b. 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. The value of the selected node can be saved to the selected item using the node link attribute or a page process. When the tree is reloaded, the tree opens to the last selected tree node.
 - c. Tooltip - Displays a tooltip when the mouse hovers over a leaf node. Options include:
 - Static Assignment - Specifies a static value to be used as the tooltip text. Substitution strings can be used in the static text, to incorporate information from the SQL query in the tooltip text. The following substitution strings can be used:
 - #VALUE# - refers to the value of ID column.
 - #TITLE# - refers to the value of Node Text column.
 - #LINK# - refers to the value of Link option.
 - Database Column - Select the column to use as the tooltip text.
 - d. Link Option - To make leaf node text a link, choose **Existing applications item** and click **Next**.

Existing Application Item makes the leaf node text a link. If you select this option, you must specify a page to link to and an existing application item to link leaf node text.
11. Confirm your selections and click **Create**.

 **Tip:**

The creation of APEX Tree regions is supported on pages using a Desktop user interface.

Creating a Tree in Page Designer

Create a tree control by providing a SQL query that specifies a hierarchical relationship by identifying an ID and parent ID column in a table or view.

The tree query you provide must utilize a `START WITH .. CONNECT BY` clause to generate the hierarchical query.

To create a tree on an existing page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. Click the **Layout** tab in the central pane.

3. In the Gallery, click the **Regions** tab. Then, right-click **Tree**, select **Add To**, and select the appropriate location.

 **Tip:**

You can also select the Tree region in the Gallery and drag it to the appropriate location in the Layout tab.

Page Designer indicates what actions are required next.

The Messages tab displays a red or yellow badge indicating messages you need to address. The Message tab displays two types of messages:

- **Errors** - Error messages display in red. Selecting an error message displays the associated attribute in red in the Property Editor. You must address errors before a page can be saved.
 - **Warnings** - Warning messages display in yellow. Selecting a warning message displays the associated attribute in yellow in the Property Editor. You must address errors before a page can be saved. Note you can save a page without addressing warning messages.
4. In either the Rendering tab or the Layout tab, select the new tree region.
The Property Editor displays attributes in the right pane. Attributes are organized in groups.
 5. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
 6. In the Property Editor, edit the attributes:
 - a. **Identification, Title** - Enter a region title. The region title only displays when it is defined in the region template.
 - b. For **Source**, select one of the following:
 - **Local Database** - Data is sourced from the local database.
 - **Remote Database** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL.
 - **Web Source** - Data is sourced from a RESTful web service defined using Web Source Modules.
 - c. **Layout** - Edit the attributes:
 - **Sequence** - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.
 - **Parent Region** - Select the parent region to which this region belongs. If a parent region is selected then this region is rendered completely inside the parent region.

- **Position** - Select the template position used to display this region. The selections available are derived from the appropriate template positions defined within the current theme, for the page's user interface.
 - d. **Appearance, Template** - Specify the template. Select a region template to define the appearance and layout of this region.
To learn more, click the **Help** tab in the central pane.
7. Edit the region Attributes:
- a. In the Rendering tab, locate the region and click **Attributes**.
The Property Editor displays the Attributes.
 - b. In the Property Editor, edit the region Attributes:
 - **Node Label Column** - Select the data source column containing the value for the node label. The value cannot contain markup.
 - **Node Value Column** - Select the data source column containing the value for the node value. This value is not displayed but added to the tree adapter node id property that can be accessed using JavaScript.
 - **Hierarchy** - Indicate whether Application Express should compute the tree hierarchy from the data or whether the data already contains all hierarchy information.
 - Computed With SQL:
 - * **Node ID Column** – Select the data source column containing the ID for a node. This is required when the hierarchy is being computed by Application Express. Typically, a primary key column is chosen here.
 - * **Parent Key Column** – Select the data source column containing the value for the parent key of a node. Required when Application Express computes the hierarchy.
 - Not Computed:
 - * **Node Status Column** – Select the data source column containing the value for the node status. Required when the **Hierarchy** attribute is set to **Not Computed**. The column must contain **0** for leaf nodes and **1** for nodes with children.
 - * **Hierarchy Level Column** – Select the data source column containing the value for the hierarchy level of the node. Required when the **Hierarchy** attribute is set to **Not Computed**.
 - **Tooltip** - Select whether tooltips are displayed, and the source for the tooltip.
 - **Link** - Specify the link to be executed when a node is clicked.
 - **Link Column** - Select the data source column containing the value for the link to be executed when a node is clicked.
 - **Icon CSS Class Column** - Select the data source column containing the value for the icon CSS class to be applied to the node.
8. Click **Save**.

Editing Tree Attributes

Developers can customize tree behavior by editing tree attributes.

By default, a tree does not have focus and node links are activated with a single click action.

To access the Tree attributes page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Select the region containing the tree in the Rendering tab or the Layout tab.
The Property Editor displays attributes in the right pane. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the tree attributes.
Edited attributes display a blue marker to the left of the attribute name until the page is saved.

Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Click **Save**.

About Incorporating JavaScript into an Application

Oracle Application Express includes multiple built-in interfaces especially designed for adding JavaScript.

Adding JavaScript to a web application is a great way to add features that mimic those found in client/server applications without sacrificing all the benefits of web deployment.

Remember that JavaScript is not appropriate for data intensive validations. For example, to verify that a name is contained within a large database table, you must pull down every record to the client, creating a huge HTML document. In general, complex operations are much better suited for server-side Application Express

validations instead of JavaScript. To learn more, see the JavaScript discussion for the appropriate control or component.

 **See Also:**

- "[Supported Shortcut Types](#)" for information about referencing a shortcut inside of a JavaScript literal string
- "[Understanding Validations](#)"

15

Adding Navigation

Create application navigation controls including tabs, navigation menus, breadcrumbs, navigation bar entries, and branches

When you build a database application, you can include different types of navigation controls. However, available navigation options depend upon the application theme. Common navigation controls include tabs, lists (including navigation menus), breadcrumbs, navigation bar entries, and branches.

- [About Using Lists as Navigation](#)
You can add navigation to your application by creating a list (or a shared collection of links).
- [Creating Tabs](#)
Navigate users between pages in an application using tabs.
- [Creating Breadcrumbs](#)
Provide users with hierarchical navigation by creating breadcrumbs. You can display a breadcrumb as a list of links or as a breadcrumb path.
- [Creating Classic Navigation Bar](#)
In older themes, developers add navigation by creating classic navigation bar entries.
- [Controlling Navigation Using Branches](#)
Add navigation to application pages by creating branches. A branch is an instruction to link to a specific page, procedure, or URL after a given page is submitted.



See Also:

["Managing Trees"](#)

About Using Lists as Navigation

You can add navigation to your application by creating a list (or a shared collection of links).

You add a list to a page by creating a region and specifying the region type as List. You control the appearance of a list through list templates. For example, you can create static list that functions as navigation using the template Vertical Unordered List with Bullets. Newer themes, such as *Universal Theme - 42*, support the creation of navigation menus and navigation bars. **Navigation menus** are lists that render at the top of the page (similar to tabs) or display as a side bar. A **navigation bar** displays with a list template in the #NAVIGATION_BAR# position on your page template.

The *Sample Database Application* includes a navigation menu and a navigation bar.

 **See Also:**

- ["Creating Lists"](#)
- ["Managing Navigation Menu"](#)
- ["Managing Navigation Bar Lists"](#)
- ["Installing and Running Sample Database Application"](#)

Creating Tabs

Navigate users between pages in an application using tabs.

 **Note:**

Tabs only appear if the associated application uses an older theme. In new themes, Tabs have been replaced with Navigation Menu.

- [About Tabs](#)
- [About the Tabs Page](#)
- [Checking for Tab Template Support](#)
- [Creating a Tab](#)
- [Editing Tabs](#)
- [Accessing Tab Reports](#)

 **See Also:**

- ["Creating Lists"](#)
- ["Managing Navigation Menu"](#)
- ["Creating Database Applications"](#)

About Tabs

Tabs are an effective way to navigate users between pages of an application. The ability to include tabs in your application depends upon your application theme. Older themes, such as such as *Theme 26 - Productivity Applications*, include tab-based navigation. Newer themes, such a *Universal Theme - 42*, include list-based navigation menus.

An application can have pages with no tabs, one level of tabs, and two levels of tabs. Standard tabs enable you to display only one level of tabs. To display two levels of tabs, you define both Parent tabs and Standard tabs.

 **Tip:**

As an alternative to tabs, you can use lists to display tab controls. List templates provide greater control over HTML generation.

App Builder includes two different types of tabs:

- **Standard tabs**

An application having only one level of tabs uses a standard tab set. A standard tab can have a one-to-one relationship with a page and is associated with a specific page and page number. You can use standard tabs to link users to a specific page.

- **Parent tabs**

Parent tabs can control the display of standard tab sets and can be current for many pages. Clicking a parent tab displays the corresponding standard tab, with the default page as the current page. Parent tabs give users another level of navigation and a context (or sense of place) within the application. You can use parent tabs to link users to a specific URL associated with a specific page.

You can group tabs into collections called a tab set. Each tab must be part of a tab set.

 **Tip:**

To see an example of an application using tabs, see the sample application, *Sample Database Application*.

 **See Also:**

["Understanding Sample Database Application"](#)

About the Tabs Page

The Tabs page describes the tabs defined in your application.

To access the Tabs page:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.


2. Under Navigation, click **Tabs**.

The Tabs page appears.


This application has 4 tabs organized into 1 tab sets, and has defined 0 parent tabs.

Tab Set: TS1


Tab: Home

 Page 1. Home - Default Page (1)


Tab: Projects

 Page 3. Projects - Default Page (1)

Tab: Employees

 Page 2. Employees - Default Page (1)

Tab: Tasks

 Page 5. Tasks - Default Page (1)

The Tab Display section provides additional information about how the current tabs or tab set are configured to display.

The Page Templates region at the bottom of the page displays a report of page templates associated with the current application. This region is hidden by default. To view it, click the greater than (>) icon to the left of Page Templates.

 **See Also:**

- ["Checking for Tab Template Support"](#)
- ["Understanding Sample Database Application"](#)

Checking for Tab Template Support

Before you can create parent and standard tabs, you must check that your default template has positions defined for both standard and parent tabs using the appropriate substitution strings. You also must make sure you do not override this template at the page-level.

To view page templates:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
2. Under Navigation, click **Tabs**.
The Tabs page appears.
3. Locate **Page Templates** at the bottom of the page. If needed, expand the Page Templates region.

| Page Templates | | | |
|---|------------------------|----------------------|-----------------------------|
| Page Template Name | Supports Standard Tabs | Supports Parent Tabs | Template Class |
| Login | No | No | Login |
| No Tabs - Left Sidebar (fixed-width / DIV based) | No | No | No Tabs with Sidebar |
| No Tabs - Left and Right Sidebar (fixed-width / DIV left and optional table-based right) | No | No | No Tabs with Sidebar |
| No Tabs - Right Sidebar (fixed-width / DIV based) | No | No | No Tabs |
| No Tabs - Right Sidebar (optional / table-based) | No | No | No Tabs |
| One Level Tabs - Left Sidebar (fixed-width / DIV based) | Yes | No | One Level Tabs with Sidebar |
| One Level Tabs - Left and Right Sidebar (fixed-width / DIV left and optional table-based right) | Yes | No | One Level Tabs with Sidebar |

See Also:

- ["Page Templates"](#)
- ["Using Themes"](#) for information about setting a default page template at the application level

Creating a Tab

An application can have pages with no tabs, one level of tabs, and two levels of tabs. Standard tabs enable you to display only one level of tabs. To display two levels of tabs, you must define both Parent tabs and Standard tabs.

To create a tab:

1. Access the Tabs page:
 - a. On the Workspace home page, click the **App Builder** icon.

The Create Parent Tab or Create Standard Tab Wizard appears.

5. Follow the on-screen instructions.



See Also:

["About Tabs"](#)

Editing Tabs

This section describes how to edit tabs. You can edit multiple tabs simultaneously. Also, for standard tabs, you can update tab properties, such as their labels and order sequence, by using the Standard Tab Tasks list.

- [Editing Multiple Tabs Simultaneously](#)
- [Managing Standard Tabs](#)

Editing Multiple Tabs Simultaneously

To edit multiple tabs simultaneously:

1. Navigate to the Tabs page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Navigation, click **Tabs**.
2. Click one of the following tabs at the top of the page:
 - **Edit Standard Tabs**
 - **Edit Parent Tabs**

A report appears.

3. To edit a specific tab, click the **Edit** icon.
4. Edit the appropriate attributes and click **Apply Changes**.

Managing Standard Tabs

The Standard Tab Task list displays on the right side of the Tabs page. You can access the links on this list to move a standard tab to different parent tab, rename a standard tab set, resequence the display order, associate pages with a tab set, create a new standard tab, or create a new standard tab set.

To access the Standard Tab Task list:

1. Navigate to the Tabs page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.

- d. Under Navigation, click **Tabs**.
2. Click **Manage Tabs**.
3. Make a selection from the Standard Tab Task list on the right side of the page:
 - **Move Highlighted Standard Tab to a Different Parent Tab**
 - **Rename Standard Tab Set**
 - **Resequence Display Order**
 - **Associate Page(s) with Selected Standard Tab**
 - **Create New Standard Tab**
 - **Create New Standard Tab Set**

Accessing Tab Reports

This section describes the Conditional Display, Utilization, and History reports.

To view the Conditional Display report:

1. Navigate to the Tabs page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Tabs**.The Tabs page appears.
2. Click the appropriate tab:
 - **Conditional Display** - The Conditional Display report displays Standard Tabs and Parent tabs that are configured to display conditionally. To filter the display, select tab type and click **Go**.
 - **Utilization** - The Utilization report lists the standard tabs used in the current application.
 - **History** - History report displays a history of changes to tab attributes for the current application.

Creating Breadcrumbs

Provide users with hierarchical navigation by creating breadcrumbs. You can display a breadcrumb as a list of links or as a breadcrumb path.

- [What Are Breadcrumbs?](#)
- [Creating a Breadcrumb While Creating a Page](#)
- [Creating a Breadcrumb Manually](#)
- [Editing Breadcrumbs](#)
- [Reparenting Breadcrumb Entries](#)
- [Deleting Unused Breadcrumb Entries](#)

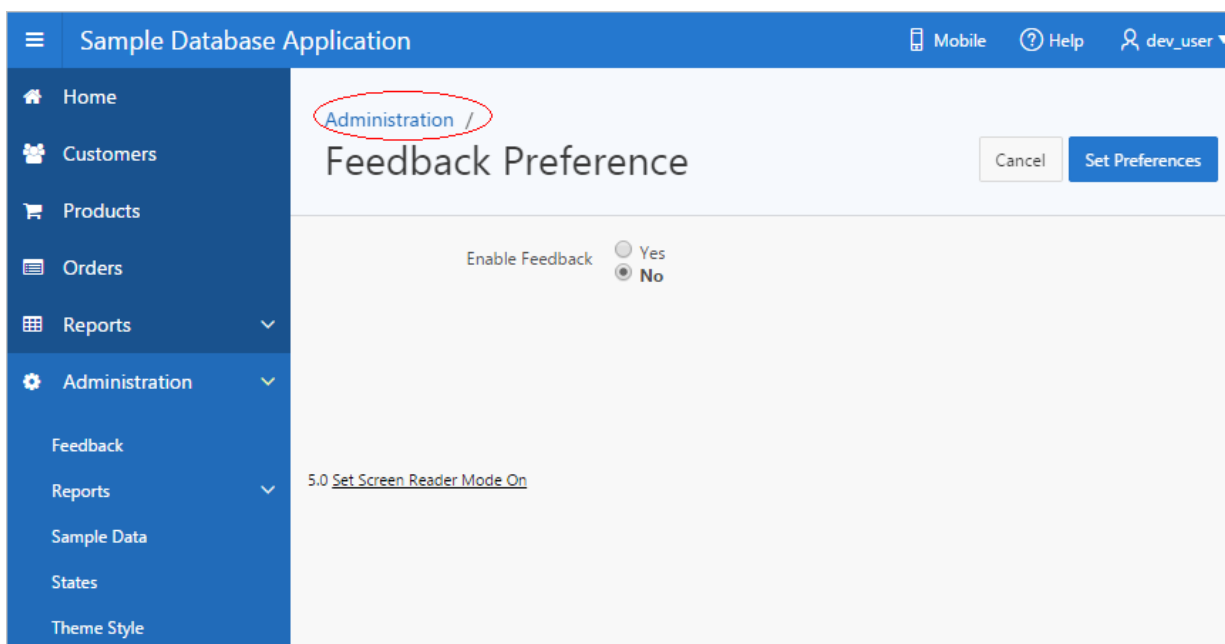
- [Accessing Breadcrumb Reports](#)

 **See Also:**

- ["Creating a Theme"](#)
- ["Breadcrumb Templates"](#)

What Are Breadcrumbs?

A breadcrumb is a hierarchical list of links that indicates where the user is within the application from a hierarchical perspective. Users can click a specific breadcrumb link to instantly view the page. You use breadcrumbs as a second level of navigation at the top of each page, complementing other user interface elements such as tabs and lists.



 **See Also:**

- ["Understanding Sample Database Application"](#)

Creating a Breadcrumb While Creating a Page

To create a breadcrumb while creating a page:

1. Run the Create Page Wizard to add a new page.

During the wizard, a Breadcrumb option appears. The actual page on which this list displays depends upon the type of page you are creating.

2. From the Breadcrumb list, select **Breadcrumb** or select an existing breadcrumb (if applicable).
3. If you select Breadcrumb:
 - a. Parent Entry - Select a parent page (if applicable) or select **No parent entry**.
 - b. Entry Name - Enter a name for the breadcrumb.
4. Follow the on-screen instructions.



See Also:

["Managing Pages in a Database Application"](#)

Creating a Breadcrumb Manually

This section describes how to create breadcrumbs manually.

- [About Creating a Breadcrumb Manually](#)
- [Creating Breadcrumbs as Shared Components](#)
- [Creating a Breadcrumb Region Using the Create Breadcrumb Wizard](#)
- [Adding a Breadcrumb Region to a Page](#)
- [Adding Entries to a Breadcrumb](#)
- [About Creating Dynamic Breadcrumbs](#)

About Creating a Breadcrumb Manually

To create breadcrumbs manually, you must add a breadcrumb to each page in your application as follows:

1. Create the breadcrumb by running the Create/Edit Breadcrumb Wizard. You can access this wizard in two ways:
 - Go the Shared Component page and select **Create Breadcrumb**.
 - In Page Designer, select the **Create** menu and then **Breadcrumb Region**.
2. Add entries to the breadcrumb.
3. Add the breadcrumb to a page by creating a region.



See Also:

["Adding Entries to a Breadcrumb"](#) and ["Adding a Breadcrumb Region to a Page"](#)

Creating Breadcrumbs as Shared Components

To create breadcrumbs from the Shared Components page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Navigation, click **Breadcrumbs**.
The Breadcrumbs page appears.
5. Click **Create Breadcrumb**.
6. Enter a name and click **Create**.
7. Add breadcrumb entries.



See Also:

["Adding Entries to a Breadcrumb."](#)

Creating a Breadcrumb Region Using the Create Breadcrumb Wizard

To create a breadcrumb region in Page Designer:

1. View the page in Page Designer.
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
Page Designer appears.
2. Locate the Page Designer Toolbar at the top of the page.
3. On Page Designer Toolbar, click the **Create** menu and select **Breadcrumb Region**.
The Create Breadcrumb Wizard appears.
4. For Region Attributes:
 - a. Region Title — Enter a title for the region.
 - b. Region Template — Select a region template.
 - c. Display Point — Identify the display point.
 - d. Sequence — Specify the sequence for this component. The sequence determines the order of evaluation.
 - e. Click **Next**.
5. For Breadcrumb:
 - a. Breadcrumb — Select the Breadcrumb to be associated with this region.
 - b. Breadcrumb Template — Identify breadcrumb template.

- c. Click **Next**.
6. For Breadcrumb Entry:
 - a. Breadcrumb — Specify the label for this breadcrumb entry.
 - b. Parent Entry— Select this breadcrumb entry's hierarchical parent..
 - c. Click **Next**.
7. Confirm your selections and click **Finish**.

Adding a Breadcrumb Region to a Page

To add a breadcrumb region to a page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

Note:

Note the Gallery at the bottom of the page. Pass the cursor over a control or component to view a tooltip that describes it.

2. In the Gallery, click the **Regions** tab. Then, right-click **Breadcrumb**, select **Add To**, and select the appropriate location.

Tip:

You can also select the Breadcrumb region in the Gallery and drag it to the appropriate location in the Layout tab.

Page Designer indicates what actions are required next.

The Messages tab displays a red or yellow badge indicating messages you need to address. The Message tab displays two types of messages:

- **Errors** - Error messages display in red. Selecting an error message displays the associated attribute in red in the Property Editor. You must address errors before a page can be saved.
 - **Warnings** - Warning messages display in yellow. Selecting a warning message displays the associated attribute in yellow in the Property Editor. You must address errors before a page can be saved. Note you can save a page without addressing warning messages.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.

- **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. In the Property Editor, edit the Breadcrumb attributes:

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

- a. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.
 - b. Layout, Sequence - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.
 - c. Layout, Position - Select the template position used to display this region. The selections available are derived from the appropriate template positions defined within the current theme, for the page's user interface.
 - d. Appearance, Template - Select a region template to define the appearance and layout of this region. To learn more, click the **Help** tab in the central pane.
 - e. Appearance, Item Display Position - Specify where page items display in relation to the main region content.
5. Click **Save**.

Repeat these procedures for each page where you would like to add breadcrumb navigation.

Adding Entries to a Breadcrumb

To add entries to a breadcrumb:

1. Navigate to the Breadcrumbs page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. Click **Shared Components**.
 - e. Under Navigation, click **Breadcrumbs**.The Breadcrumbs page appears.
2. Select a breadcrumb to which to add entries.
3. Click **Create Breadcrumb Entry**.
4. Under Breadcrumb, select the page where this breadcrumb entry displays.
5. Under Entry:
 - a. Sequence - Indicate the order in which breadcrumb entries appear. The sequence determines the order of evaluation.
 - b. Parent Entry - Identify the parent of this entry. Any given breadcrumb should have only one root node, identified as an item with no parent ID.

- c. Short Name - Identify the short name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.
 - d. Long Name - Identify the long name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.
 6. Under Target, specify the target location:
 - a. Target is a - Select **Page in this Application**.
 - b. Page - Specify the target page number.

To reset pagination for that page, select **reset pagination for this page**.
 - c. Request - Enter text for which you would like to set the built-in application item called `REQUEST`. This is also the item that is set with the name of a button that was clicked.
 - d. Clear Cache - Enter the page numbers (separated by commas) for those pages for which you would like to clear the user's session state.
 - e. To set session state (that is, give a listed item a value):
 - Set these items - Enter the list of application item names (separated by commas) for which you would like to set the session state (give the listed item a value).
 - With these values - Enter a comma delimited list of values for the items specified in Set these items. You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`).

You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, `\1234,56\`).
7. For Conditions:
 - a. Condition Type - Select a condition type from the list. This condition must be met in order for this component to be rendered or processed.
 - b. Enter an expression in the fields provided.
8. For Authorization, optionally select an authorization scheme. This authorization scheme must evaluate to TRUE in order for this component to be rendered or otherwise processed.
9. For Configuration, select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.
10. Click **Create Breadcrumb Entry**.

Repeat these procedures for each breadcrumb entry you create.

About Creating Dynamic Breadcrumbs

To give users more exact context, you can include session state in breadcrumbs, making your breadcrumbs dynamic. For example, suppose a page in your application displays a list of orders for a particular company and you want to include the following breadcrumb:

Home > Orders > Orders for ACME Inc

In this example, `ACME Inc` not only indicates the page a user is on but also the navigation path. The Application Express engine stores the value of `ACME Inc.` in session state.

To create this type of dynamic menu, you must include a reference to a session state item in the breadcrumb's short name or long name. For example:

```
&COMPANY_NAME.
```

Editing Breadcrumbs

Once you create a breadcrumb, you can edit it on the Breadcrumbs page.

To edit a breadcrumb:

1. Navigate to the Breadcrumbs page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, select **Breadcrumbs**.The Breadcrumbs page appears.
2. Select a breadcrumb.
3. Select a breadcrumb entry.
The Create/Edit page appears.
4. Edit the appropriate attributes.
To learn more about a specific item, see field-level Help.
5. Optional) In the Breadcrumb Entry list on the right side of the page, you can select the **Synchronize Breadcrumb With Page Name and Title** option.
Selecting this option makes changing the name of a page and breadcrumb a one-step process. The information you provide for breadcrumb names is used to update the referenced page name and title.
6. Click **Apply Changes**.

See Also:

- ["Viewing Field-Level Help"](#)
- ["Accessing Breadcrumb Reports"](#)

Reparenting Breadcrumb Entries

You can select a new parent for selected breadcrumb entries on the Reparent Entries page.

To reparent breadcrumb entries:

1. Navigate to the Breadcrumbs page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, select **Breadcrumbs**.The Breadcrumbs page appears.
2. Select a breadcrumb.
The Entries page appears.
3. From the Tasks list, click **Reparent Entries within this Breadcrumb**.
The Reparent Entries page appears.
4. Use the navigation bar to edit or filter the view:
 - Breadcrumb - Identify the breadcrumb you want to edit and click **Go**.
 - Start With - Make a selection to restrict your view to a subset of the breadcrumb hierarchy and click **Go**.
5. From Reparent to, select the new parent.
6. Select the breadcrumbs entries you want to move and click **Reparent Checked Entries**.

Deleting Unused Breadcrumb Entries

To delete unused breadcrumb entries:

1. Navigate to the Breadcrumbs page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, select **Breadcrumbs**.The Breadcrumbs page appears.
2. Select a breadcrumb.
The Breadcrumb Entries page appears.
3. From the Tasks list, click **Delete Unused Breadcrumb Entries**.
The Delete page appears.
4. Select the entries to remove and click **Delete Checked**.

Accessing Breadcrumb Reports

You can view the Breadcrumb Hierarchy, Grid Edit Breadcrumb, Breadcrumb Exceptions, Breadcrumb Utilization and Breadcrumb History reports by clicking the appropriate tab at the top of the Breadcrumbs page.

 **Note:**

These reports only appear after you create a breadcrumb.

To view Breadcrumb reports:

1. Navigate to the Breadcrumbs page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, select **Breadcrumbs**.
The Breadcrumbs page appears.
2. Click the appropriate tab:
 - **Hierarchy** - Click **Hierarchy** to view breadcrumbs by page and hierarchy. Click the page number to link to a page.
 - **Grid Edit** - Click **Grid Edit** to edit all breadcrumb names at once. Click the page number to link to a page.
 - **Exceptions** - Click **Exceptions** to view the Exceptions page.
 - **Utilization** - Click **Utilization** to access the Breadcrumb Utilization report. This report lists breadcrumbs by page. Click the page number to go to a specific page.
 - **History** - Click **History** to view the Breadcrumb History report. This report lists recent changes to breadcrumbs.

You can change the appearance of the page by using the Search bar at the top of the page.

 **See Also:**

["Customizing Interactive Reports in a Running Application"](#)

Creating Classic Navigation Bar

In older themes, developers add navigation by creating classic navigation bar entries.

 **Tip:**

Applications using the *Universal Theme - 42* theme provide navigation support through the creation of navigation menus.

- [About Classic Navigation Bar Entries](#)
- [Creating a Classic Navigation Bar Entry](#)
- [Copying a Classic Navigation Bar Entry](#)
- [Editing a Classic Navigation Bar Entry](#)
- [Editing Multiple Classic Navigation Bar Entries Simultaneously](#)
- [Accessing Classic Navigation Bar Entry Reports](#)



See Also:

- ["Creating Custom Themes"](#)
- ["Managing Navigation Menus"](#)

About Classic Navigation Bar Entries

For applications using older themes, navigation bar entries offer an easy way to move users between application pages. The associated page template determines the location of a navigation bar. A navigation bar entry can be an image, text, or an image with text beneath it. You must supply the images and text to use in the navigation bar entries.

Navigation bars are different from other shared components in that you do not need to reference them on a page-by-page basis. If your page template includes the #NAVIGATION_BAR# substitution string, the Application Express engine automatically includes any defined navigation bars when it renders the page.

Creating a Classic Navigation Bar Entry

Before adding a navigation bar, you must create entries for the navigation bar. You can create a navigation bar entry from scratch or by copying an existing entry.

- [Creating a Classic Navigation Bar Entry from Scratch](#)
- [Creating a Classic Navigation Bar Entry for Feedback](#)

Creating a Classic Navigation Bar Entry from Scratch

To create a navigation bar entry from scratch:

1. Navigate to the Navigation Bar Entries page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click **Create**.

The Create Navigation Bar Entry Wizard appears.

3. For Method, select **From Scratch**.
4. For Type, select **Navigation to URL**.
5. For Attributes, specify the following:
 - a. Sequence - Specify the order of evaluation for this component.

 **Tip:**

To review the existing entries, click the **Existing Navigation Bar Entries** link at the bottom of the page.

- b. Entry Label - Enter display text for this navigation bar entry.
 - c. Icon Image Name - Enter an image name. For naming conventions, see field-level Help.
 - d. Image ALT - Enter ALT text for navigation icons that are images. If you do not specify an image name, then this text displays.
 - e. Image Height - Define the height of the image in pixels.
 - f. Width - Defines the width of the image.
 - g. Click **Next**.
 6. For Target, specify the following
 - a. If the target location is a URL, specify:
 - Target is a - Select **URL**.
 - URL Target - Enter a URL. For example:
`http://www.yahoo.com`
 - b. If the target location is a page, specify:
 - Target is a - Select **Page in this Application**.
 - Page - Specify the target page number.
To reset pagination, select **reset pagination for this page**.
Select **Printer Friendly** to display the target page using the application's Printer Friendly template. Printer friendly templates optimize a page for printing.
 - Request - Enter text that defines the built-in application item called `REQUEST`.
 - Clear Cache - Enter the page numbers where you would like to clear the user's session state. Separate multiple entries with commas.
 - c. To set session state (that is, give a listed item a value):
 - Set these items - Enter the list of application item names for which you would like to set session state. Separate multiple items with commas. Setting the session state gives the listed item a value.
 - With these values - Enter a comma-delimited list of values for the items your specified.
You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL

cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, \1234,56\).

- d. Click **Next**.
7. To set a condition for displaying the navigation bar entry, select a condition type.
8. Click **Create**.

 **Tip:**

In order for a navigation bar to appear, your page template must include the #NAVIGATION_BAR# substitution string. To learn more, see "[Page Templates](#)."

 **See Also:**

- "[Creating a Classic Navigation Bar Entry for Feedback](#)"
- "[Viewing Field-Level Help](#)"
- "[Editing Themes](#)"
- "[Optimizing a Page for Printing](#)"

Creating a Classic Navigation Bar Entry for Feedback

Feedback is the process of gathering real-time comments, enhancement requests, and bugs from your application users. To add a feedback page to an application, you add a Feedback page.

To create a navigation bar entry for feedback:

1. Navigate to the Navigation Bar Entries page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click **Create**.

The Create Navigation Bar Entry Wizard appears. Each wizard page displays a series of blocks on the left, representing the sequence of steps. The highlighted block indicates the step you are performing.
3. For Method, select **From Scratch**.
4. For Type, select **Feedback**.
5. For Feedback:
 - a. Feedback Page - Select your feedback page.

- c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Classic Navigation Bar Entries**.
 2. Select a navigation bar entry.
The Edit page appears.
 3. Edit the appropriate attributes.
 4. To learn more about a specific item on a page, see field-level Help.
 5. Click **Apply Changes**.



See Also:

["Viewing Field-Level Help"](#)

Editing Multiple Classic Navigation Bar Entries Simultaneously

To edit multiple navigation bar entries simultaneously:

1. Navigate to the Lists page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder**.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Select a list.
3. Click **Grid Edit** at the top of the page.
4. Edit the appropriate attributes and click **Apply Changes**.

Accessing Classic Navigation Bar Entry Reports

You can view the Navigation Bar Entry Subscription and Navigation Bar Entry History reports by clicking the appropriate tab at the top of the Navigation Bar Entries page.



Note:

The Subscription and History buttons only appear after you create a navigation bar.

1. Navigate to the Navigation Bar Entries page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.

- c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click the appropriate tab:
 - **Subscription** - Click **Subscription** to access the Subscribed NavBars report. This report displays subscribed navigation bar entries in your application.
 - **History** - Click **History** to view the Navigation Bar History report. This report lists recent changes to navigation bars.

Controlling Navigation Using Branches

Add navigation to application pages by creating branches. A branch is an instruction to link to a specific page, procedure, or URL after a given page is submitted.

- [Creating a Branch](#)
- [About Branching Conditionally](#)



See Also:

["Accessing Page Specific Utilities"](#)

Creating a Branch

To create a branch:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In either the Rendering or Processing tab, right-click **Branches** and select **Create Branch**.

The Property Editor displays the Branch attributes.



Tip:

The Messages tab displays a red or yellow badge to identify messages you need to address. Selecting a message displays the associated attribute in the Property Editor. You must address red error message before you can save.

- a. Identification, Name - Specify the name of the branch for easy identification by developers.

- b. Execution Options:
 - i. Sequence - Specify the sequence for this component. The sequence determines the order of execution.
 - ii. Point - Select the point in the processing at which this branch action is considered. Options include:
 - **After Submit** (previously named **Before Computation**)
 - **Validating** (previously named **Before Validation**)
 - **Processing** (previously named **Before Processing**)
 - **After Processing** (previously named **After Processing**)
 - **Before Header** (previously named **Before Header**)
 - c. Behavior, Type - Select a page branch type. To learn more, click the **Help** tab in the central pane.

The attributes that display in the Property Editor depend on the branch type you select.

If Type is **Page or URL (Redirect)**, click **Target** and edit the attributes in the Link Builder Target dialog.
 - d. Server-Side Condition:
 - i. When Button Pressed - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.
 - ii. Condition Type - Select a condition type from the list that must be met in order for this component to be rendered or processed. To learn more, click the **Help** tab in the central pane.
3. Edit the Branch attributes:

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Click **Save**.

 **See Also:**

- ["About the When Button Pressed Attribute"](#)
- ["Creating Pre-Rendering Branches, Computations, and Processes"](#)
- ["Creating Page Submission Branches, Validations, Computations, and Processes"](#)

About Branching Conditionally

Like other controls, branches can be made conditional. To a branch conditional, view the branch attributes in the Property Editor. Under Server-side Condition, edit the appropriate attributes.

16

Managing Computations, Validations, and Processes

Enhance your application by including computations, validations, and processes.

Computations enable you to assign values to items. **Validations** are edit checks on specific items, pages, or columns. A **page process** performs an action at a specified point during the rendering or submission of the page. **Application processes** are blocks of PL/SQL logic that are set to run at specific points using processes from multiple pages of an application.

- [Understanding Page Computations](#)
Use computations to assign a value to an identified item when a page is submitted or displayed.
- [Understanding Validations](#)
Use validations to create edit checks on specific items, pages, or columns.
- [Understanding Page Processes](#)
Create **page processes** to perform an action at a specified point during the rendering or submission of the page.
- [About Branches](#)
A branch is an instruction to go to a specific page, procedure, or URL. For example, you can branch from page 1 to page 2 after page 1 is submitted. When you create a branch, you specify a Branch Point and Branch Type.
- [Understanding Application Processes](#)
Create an application process to run a block of PL/SQL logic at a specific point from multiple pages of an application. By default, application processes execute at the same point for every page in the application. However, you can apply conditions for specific pages to control when the process executes.
- [Understanding Application Computations](#)
Create an application computation to set the value of a single page or application-level item. Application computations run at the same point across multiple pages in an application.
- [Using the Attribute Dictionary](#)
Use the Manage Attribute Dictionary page to update values on the current page with those in the Attribute Dictionary, or to update the Attribute Dictionary with the values on the current page.

Understanding Page Computations

Use computations to assign a value to an identified item when a page is submitted or displayed.

- [About Page Computations](#)
- [Creating a Page Computation](#)

- [Editing Page Computation Attributes](#)
- [Editing the Computation Point and Source](#)
- [Computation Points and Computation Syntax Example](#)



See Also:

["Understanding Application Computations"](#)

About Page Computations

A page computation assigns a value to an identified item when a page is displayed or submitted (rendered and processed). You create a page computation by running the Create Page Computation Wizard. For each computation, specify the item for which you are creating the computation and a computation type.

You can also use application-level computations to assign values to items. Most page-level computations populate page items. In contrast, most application-level computations populate application items.

Creating a Page Computation

To create a page computation:

1. Navigate to the appropriate page in Page Designer.
2. In the Rendering tab, expand the **Pre-Rendering** or **Post-Rendering** node and locate where the computation should be.
3. Right-click the location and select **Create Computation**.

Page Designer appears.

In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.



Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Edit the following attributes:
 - a. Identification, Item Name - Enter the page or application item populated by this computation.
 - b. Execution Options, Sequence - Specify the sequence for this computation. The sequence determines the order of execution.
 - c. Execution Options, Point - Select at what point in page rendering or processing this computation is executed. The most typical computation point is **After Submit**.

- d. Computation, Type - Select a computation type. To view a list of options, see Help.
5. Click **Save**.

 **See Also:**

- ["Viewing a Page in Page Designer"](#)
- ["Creating Pre-Rendering Branches, Computations, and Processes"](#)
- ["Creating Page Submission Branches, Validations, Computations, and Processes"](#)

Editing Page Computation Attributes

Once you create a computation, you can edit it on the Edit Page Computation page.

To edit a page computation:

1. View the page in Page Designer.
2. In the Rendering tab, expand the Pre-Rendering or Post-Rendering node and select the computation.

Page Designer appears.

In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

3. In the Property Editor, edit the appropriate attributes.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

4. Click **Save**.

 **See Also:**

- ["Viewing a Page in Page Designer."](#)

Editing the Computation Point and Source

You control when a computation executes by editing the **Sequence** and **Point** attributes. The **Sequence** attribute determines the order of evaluation. The **Point** attribute determines the point in page rendering or processing at which the computation executes. The most typical computation point is After Submit.

To edit a computation point and source:

1. View the page in Page Designer.
2. In the Rendering tab, expand the Pre-Rendering or Post-Rendering node and select the computation.

Page Designer appears.

In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

3. In the Property Editor, edit the following attributes.
 - a. Execution Options, Sequence - Specify the sequence for this component. The sequence determines the order of execution. If two components have the same sequence value, then they may be executed in a different order when the application is exported and imported into another environment.
 - b. Execution Options, Point - Select at what point in page rendering or processing this computation is executed.

For example, selecting **On New Instance** executes the computation when a new session (or instance) is generated. The most typical computation point is **After Submit**.

- c. Computation - Define expression or query to compute an item's value.
- d. Error, Error Message - Enter an error message that displays if this computation fails.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

4. Click **Save**.

 **See Also:**

["Viewing a Page in Page Designer"](#)

Computation Points and Computation Syntax Example

A good example of using computations can be illustrated by a page containing form fields for entering phone numbers. In this example, the phone number is stored in one database column; however, the data entry form breaks the phone number into three components: area code, prefix, and line number. In this example, the page items are called `P10_AREA_CODE`, `P10_PREFIX`, and `P10_LINE_NUMBER`.

Next, suppose you need to combine the values stored in these items into a single string. You could accomplish this by using an After Submit computation and store the combined values in an item called `P10_PHONE_NUMBER`.

To create a computation to store the combined values of P10_AREA_CODE, P10_PREFIX, and P10_LINE_NUMBER in new items:

1. Navigate to the appropriate page.
2. Create an item named P10_PHONE_NUMBER to store the combined values of P10_AREA_CODE, P10_PREFIX, and P10_LINE_NUMBER.
3. Create the computation:
 - a. In the Rendering tab, expand the **Pre-Rendering**.
 - b. Select a location, and then right-click and select **Create Computation**.

Page Designer appears.

In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. In the Property Editor, edit the following attributes:
 - a. Identification, Item Name - Select P10_PHONE_NUMBER.
 - b. Execution Options, Sequence - Specify the sequence for this computation. The sequence determines the order of execution.
 - c. Execution Options, Point - Select at what point in page rendering or processing this computation is executed. The most typical computation point is **After Submit**.
5. For Computation Type, you can create the following computation types:
 - a. Static Value:
 - Computation, Type - Select **Static Value** .
 - Computation, Static Value - Enter the following computation:
 - b. PL/SQL Function Body:
 - Computation, Type - Select **PL/SQL Function Body** .
 - Computation, PL/SQL Function Body - Enter the computation:

```
(&P10_AREA_CODE.) &P10_PREFIX.-&P10_LINE_NUMBER.
```

```
DECLARE
l_return_value VARCHAR2(300) DEFAULT NULL;
BEGIN
  l_return_value :=
  '('||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER;
RETURN l_return_value;
END;
```

- c. SQL Query (return colon separated value):
- Computation, Type - Select **SQL Query (return colon separated value)** .
 - Computation Type, SQL Query - Enter the computation:

```
SELECT
'('||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER
FROM DUAL
```

- d. PLSQL Expression:
- Computation, Type - Select **PLSQL Expression** .
 - Computation Type, PLSQL Expression - Enter the computation:

```
'('||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER
```

6. Click **Save**.



See Also:

["About the Differences Between Page Items and Application Items"](#)

Understanding Validations

Use validations to create edit checks on specific items, pages, or columns.

- [About Validations](#)
- [Creating a Validation](#)
- [Editing a Validation](#)
- [About Determining When Validations Execute](#)
- [About Defining How Validation Error Messages Display](#)
- [Processing Validations Conditionally](#)
- [About Altering the Go to Error Link](#)
- [What Happens When a Validation Fails?](#)

About Validations

A validation is an edit check. Validations specific to a single item are **page item validations**. Validations that apply to an entire page are **page validations**. Validations for tabular forms specific to a single column are **column level validations**. Validations for tabular forms that do not apply to a single column are **tabular form row validations**.

You can define a validation declaratively by selecting a validation method. You enter the actual validation edit check in the Validation Messages field. Be aware that if a validation fails, subsequent page processes or computations do not occur. Also

remember that the validation you enter must be consistent with the validation type you selected.

Creating a Validation

To create a validation in Page Designer:

 **Note:**

Text entered for validations may not exceed 3,950 characters.

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

2. Click the **Processing** tab in the left pane.
3. Right-click **Validations** and select **Create Validation**.

Validation attributes display in the Property Editor. Attributes are organized in groups.

4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Edit the following attributes:

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- a. Identification, Name - Enter the name of the validation.
- b. Execution Options, Sequence - Enter the sequence for this validation. The sequence determines the order of execution.
- c. Validation, Editable Region - Select the associated region.
- d. Validation, Type - Select the type of equality to be tested for this validation. Depending on your selection, one or more additional attributes are required to fully define this validation.

If the validation passes the equality test, or evaluates to TRUE, then the validation error message does not display. Validation error messages display

when the validation fails the equality test, or evaluates to FALSE, or a non-empty text string is returned.

- e. Validation, Always Execute - Specify whether this validation always executes.

If set to **Yes**, this validation is always evaluated, irrespective of the Execute Validations setting defined against the button that submitted the page. If set to **No**, this validation is only evaluated if the triggering button has an Execute Validations setting of **Yes**.

- f. Error, Error Message - Enter text for the error message.
- g. Error, Display Location - Select where the error message displays for this validation.

Validation error messages display on a separate error page, or inline with the existing page. Inline error messages display underneath the Associated Item label and/or in a Notification area, defined as part of the page template.

- h. Error, Associated Item/Column - Select the item or column where this validation error message displays.

If you select an item and the Error message display location includes **Inline with Field**, the error message displays underneath the label of the specified item. If the Error message display location does not include **Inline with Field**, this attribute has no impact on the display of the error message.

- 6. Click **Save**.



See Also:

- ["Creating Pre-Rendering Branches, Computations, and Processes"](#)
- ["Creating Page Submission Branches, Validations, Computations, and Processes"](#)

Editing a Validation

To edit a validation in Page Designer:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Click the **Processing** tab in the left pane.
3. Expand **Validating** and then **Validations** and select the validation.
Property Editor displays Validation attributes. Attributes are organized in groups.
4. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.

- **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. In the Property Editor, edit the appropriate attributes.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

6. Click **Save**.

About Determining When Validations Execute

When creating a validation, the **Always Execute** attribute determines if validations execute when a page is submitted. Use this attribute for buttons or certain page items that submit a page. The Always Execute attribute has two options:

- **Yes** - The validation always executes independent of validation settings for buttons or items on a page.

For example, select **Yes** if your validation is a security check which determines if the current user is allowed to modify or delete the current record. This ensures the validation executes for any operation and prevents unprivileged users from modifying data.

- **No** - The validation only executes if the button or item used to submit the page has the **Execute Validations** attributes set to **Yes**.

For example, it is not useful to verify that an item be numeric and display an error message if the user is simply deleting a record. In that example, the validations should only execute when the user clicks the Create or Apply Changes buttons.

To accomplish this behavior:

- For the validation, set Always Execute to **No**
- For the Create and Apply Changes buttons, set Execute Validations to **Yes**.
- For the Delete button, set Execute Validations to **No** to avoid unnecessary validations.

 **See Also:**

["Editing a Validation"](#)

About Defining How Validation Error Messages Display

You can choose to have validation error messages display inline (that is, on the page where the validation is performed) or on a separate error page.

You define how a validation error message displays by configuring the following validation attributes:

1. Error Message - Enter the error message text in this field.
2. Display Location - Select where the error message displays for this validation. Validation error messages display on a separate error page, or inline with the existing page. Inline error messages display underneath the Associated Item label and/or in a Notification area, defined as part of the page template.

 **Tip:**

From Component View, this attribute is named **Error Message display location**.

Options include:

- **Inline with Field and in Notification** - Error displays inline within the field label and within a notification area (defined as part of the page template).
 - **Inline with Field** - Error displays inline within the field label.
 - **Inline in Notification** - Error displays inline within a notification area (defined as part of the page template).
 - **On Error Page** - Displays the error on a separate error page. To create a hard error that stops all processing (including validations), you must display the error on an error page.
3. Associated Item (or Column) - Select the item (or column) where this validation error message displays.

If you select an item and the Error message display location includes **Inline with Field**, the error message displays underneath the label of the specified item. If the Error message display location does not include **Inline with Field**, this attribute has no impact on the display of the error message.

 **See Also:**

["Editing a Validation"](#)

Processing Validations Conditionally

You can control when and if a validation is performed by configuring the following Conditions attributes:

1. When Button Pressed - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.

If no button is pressed, other conditions are evaluated before performing the validation.

2. Execution Scope - Applicable only to tabular forms. Specify the execution scope:
 - **For Created and Modified Rows** - The validation executes only for created and modified rows in your tabular form.

- **All Submitted Rows** - The validation executes for every submitted row.
3. **Condition Type** - Select a condition type from the list that must be met in order for this component to be rendered or processed. To view a list of options, click the **Help** tab.

 **See Also:**

- ["Editing a Validation"](#)
- ["Understanding Conditional Rendering and Processing"](#)

About Altering the Go to Error Link

If you define an associated item, Oracle Application Express renders a **Go to Error** link in the Notification display location next to the corresponding error message. When clicked, the user's focus is set to the associated item.

You can alter the **Go to Error** text to change the default text or to translate it into a language that is not translated by default with Oracle Application Express. In either case, define a text message called `APEX.GO_TO_ERROR` with the text and language of your choice. Additionally, you can disable this functionality and remove the **Go to Error** link by setting the text of the `APEX.GO_TO_ERROR` text message to just a single space.

 **See Also:**

- ["Translating Messages Used Internally by Oracle Application Express"](#)

What Happens When a Validation Fails?

If a page is submitted and some of the validations fail, Oracle Application Express redisplay the existing page with all inline validation errors. While displaying inline errors, Oracle Application Express does not execute computations, application processes, or page processes (for example, Automated Row Fetch) which are defined to execute during Page Rendering (in other words, all "On Load %" display points). The one exception is if the computations and processes use the condition type Inline Validation Errors Displayed.

The reason for skipping these computations and processes is that any of them could potentially alter the data entered by the user. For example, an Automated Row Fetch process would fetch and overwrite the modified data with the values from the database when the user actually wants to get the entered data in order to fix the validation error.

If a computation is skipped, the following displays in the debug output:

```
Do not perform computation because inline validation errors found.
```

If an application or page process is skipped, the following displays in the debug output:

Skip because inline validation errors found.

Understanding Page Processes

Create **page processes** to perform an action at a specified point during the rendering or submission of the page.

- [About Page Processes](#)
- [Creating Page Processes](#)
- [Editing Page Process Attributes](#)
- [Controlling When a Page Process Executes](#)
- [Creating Page Process Success and Error Messages](#)
- [Making a Page Process Conditional](#)

About Page Processes

A page process performs an action at a specified point during the rendering or submission of the page. For example, you can create a page process to execute logic or to make a call to the Application Express engine. A page process is a unit of logic that runs when a specific event occurs, such as loading or submitting a page.

From a functional perspective, there is no difference between page-level and application-level processes. The difference between these two process types is where the process is defined, that is at the page-level or at the application level.



See Also:

- ["Understanding Application Processes"](#)
- ["Accessing Page Specific Utilities"](#)

Creating Page Processes

To create a page process:

1. View the page in Page Designer.
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.

Process attributes display in the Property Editor. Attributes are organized in groups.

**Tip:**

The Messages tab displays a red or yellow badge to identify messages you need to address. Selecting a message displays the associated attribute in the Property Editor. You must address red error message before you can save.

3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

**Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

4. Under Identification:
 - a. Name - Enter the name of the process.
 - b. Type - Specify which process type to execute. The Type you select determines what Setting attributes display.
5. Under Settings, edit the appropriate attributes.
6. Under Execution Options:
 - a. Sequence - Specify the sequence for this computation. The sequence determines the order of execution.

If two components have the same sequence value, then they may be executed in a different order when the application is exported and imported into another environment.
 - b. Point - Select the point at which this process is executed.
 - c. Run Process - Select the frequency that this process is executed.
7. Under Success Message:
 - a. Error Message - Enter the success message for this process.

If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed.
8. Under Error:
 - a. Error Message - Enter the error message for this process.

This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays.
 - b. Display Location - Select where this process error message displays, when the process fails to complete successfully.
9. Click **Save**.

 **See Also:**

- "Creating Pre-Rendering Branches, Computations, and Processes"
- "Creating Page Submission Branches, Validations, Computations, and Processes"

Editing Page Process Attributes

To edit an existing page process:

1. View the page in Page Designer.
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
Page Designer appears.
2. In either the Rendering or Processing tab, expand the nodes to locate and select the process.

Process attributes display in the Property Editor. Attributes are organized in groups.

3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes:

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save**.

Controlling When a Page Process Executes

To control when a process executes, edit the following attributes:

1. View the process attributes in the Property Editor.
2. Edit the following attributes:
 - a. Execution Options, Sequence - Specify the sequence for this component. The sequence determines the order of execution.

- b. Execution Options, Point - Select the point in the processing at which this branch action is considered.
3. Click **Save**.



See Also:

["Editing Page Process Attributes"](#)

Creating Page Process Success and Error Messages

You can define both success and error messages that display when a process runs. How an error message displays depends upon the processing point. For processes with a processing point of **On Submit - Before Computation and Validations** or **On Submit - After Computations and Validations**, you can specify whether the error message displays inline on the current page (the default) or on a separate page. For all other page processes, error messages display on a separate error page.

To create a page process error or success message:

1. View the process attributes in the Property Editor.
2. Edit the following attributes:
 - a. **Success Message, Success Message** - Enter the success message for this process. If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed. If you are branching to another page via a URL redirect, you may need to check the preserve success message attribute.
 - b. **Error, Error Message** - Enter the error message for this process. This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays.
 - c. **Error, Display Location** - Select where this process error message displays, when the process fails to complete successfully.

Option include:

- **Inline in Notification** - The process error message displays in the Notification area (defined as part of the page template).
 - **On Error Page** - Error displays on a separate error page.
3. Click **Save**.



See Also:

- ["Editing Page Process Attributes"](#)
- ["Page Templates"](#)

Making a Page Process Conditional

To create a conditional page process, edit the following attributes:

1. View the process attributes in the Property Editor.
2. Under Server-side Condition, edit the following attributes:
 - a. When Button Pressed - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.
 - b. Execution Scope - Applicable only to editable interactive grids. Specify the execution scope:
 - **For Created and Modified Rows** - The validation executes only for created and modified rows in your tabular form.
 - **All Submitted Rows** - The validation executes for every submitted row.
 - c. Condition Type - Select a condition type from the list that must be met in order for this component to be rendered or processed.
3. Click **Save**.



See Also:

["Editing Page Process Attributes"](#)

About Branches

A branch is an instruction to go to a specific page, procedure, or URL. For example, you can branch from page 1 to page 2 after page 1 is submitted. When you create a branch, you specify a Branch Point and Branch Type.



See Also:

["Controlling Navigation Using Branches"](#)

Understanding Application Processes

Create an application process to run a block of PL/SQL logic at a specific point from multiple pages of an application. By default, application processes execute at the same point for every page in the application. However, you can apply conditions for specific pages to control when the process executes.

 **Tip:**

To learn more about execution behavior in the case of a validation error, see "[What Happens When a Validation Fails?](#)"

- [On Demand Application Processes](#)
- [About Application Processes that Execute On New Instance](#)
- [About Running an On Demand Process from a Page Request](#)
- [Application Process Example](#)
- [Creating an Application Process](#)
- [Editing Application Process Attributes](#)
- [Creating Application Process Error Messages](#)
- [Viewing the Application Processes History Report](#)

 **See Also:**

["Understanding Page Processes"](#)

On Demand Application Processes

An **On Demand** process is special type of application process which has a Process Point of On Demand, is of type PL/SQL, and executes when called from a page-level On Demand process or from an Ajax call from the browser. On Demand processes are useful when you have PL/SQL logic that you would like to run from different execution points across multiple pages.

On Demand processes should typically be created on a page and not at the application-level. On Demand processes created at the application-level are created with an initial authorization scheme of **Must Not Be Public User**. This prohibits the processes being invoked from users in unauthenticated sessions.

 **See Also:**

["Creating Page Processes"](#)

About Application Processes that Execute On New Instance

Typically an application process runs at the same point across multiple pages in an application. Processes having a Process Point of **On New Instance** are the exception. These types of processes are useful when you only need to retrieve information once within a user's session. For example, if the application items do not depend on the

logged in user, you can initialize them using the Process Point **On New Instance**. To look up information that depends on the user, you can also use **After Authentication**.

If you configure applications to share the same session by setting the same cookie name in the authentication scheme, **On New Instance** and **After Login**, the application processes fire whenever Oracle Application Express first processes a request for the application (that is, On New Instance), or processes a request of an authenticated user for the application (that is, **After Authentication**).

About Running an On Demand Process from a Page Request

You can have a page request run an On Demand process by using the following syntax:

```
f?p=application_id:page_id:session:APPLICATION_PROCESS=process_id
```

Where:

- *application_id* is the application ID or alphanumeric alias
- *page_id* is the page number or alphanumeric alias
- *session* is the session ID
- *APPLICATION_PROCESS=process_id* is the keyword *APPLICATION_PROCESS=* followed by either the process ID or an alphanumeric name of an application-level process having a Process Point of On Demand

When you use this syntax, the Application Express engine recognizes the request and processes it using the following rules:

- The page number in the URL can be the current page number or alias. A page number or alias is required in the request only as a syntactic placeholder because no specific page is accessed for this type of request.
- The process authorization scheme, the application's authorization scheme, and the process conditions are supported.
- Session state (that is, item names and values) may be set in the URL, but clear cache options are ignored.
- Any failures of authentication, authorization, or process conditions do not result in visible error messages or other indicators of such failures and most often result in a blank page being displayed. Note that if you are logged in to App Builder as a developer, an error messages displays.
- Specifying the process by name locates the first process with the specified (case-preserved) name.



See Also:

["Clearing Session State"](#)

Application Process Example

A shopping cart application is a good example of when you might use an application process. For example, to display the contents of a user's shopping cart with each page view, you create a region on page zero of your application that displays the values of the application-level items `TOTAL_CART_ITEMS` and `TOTAL_PURCHASE_PRICE`.

Instead of writing a process for each page to set the values of `TOTAL_CART_ITEMS` and `TOTAL_PURCHASE_PRICE`, you could write an application process of type **On Load: Before Header** to compute these values. Then, the Application Express engine would execute the process on each page as it renders the application. As a result, each page, would display the most current values for `TOTAL_CART_ITEMS` and `TOTAL_PURCHASE_PRICE`.



See Also:

["Creating a Global Page to Display Components on Every Page"](#)

Creating an Application Process

To create an application process:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Processes**.
The Application Processes page appears.
3. Click **Create**.
4. For Identification:
 - a. Name - Enter a name for the application process.
 - b. Sequence - Specify the sequence number for this process. The sequence number determines the order in which the process is evaluated relative to other processes that execute at the same point.
 - c. Point - Identify the point during the processing of each page in the application that this process would run. Note the option, **On Demand**, only runs when requested by a page process.
 - d. Click **Next**.
5. For Source:
 - a. PL/SQL code - Enter the PL/SQL code of your application process. The code will automatically be wrapped with a `Begin` and `End;`.
 - b. Error Message - Enter the error message that displays if the process raises an error.

- b. Error message display location - Applies to processes with a processing point of `On Submit - Before Computation and Validations` or `On Submit - After Computations and Validations`, you can specify where the error message displays.

Select one of the following:

- **Inline in Notification** - Error displays inline on the page in the "notification" area (defined as part of the page template).
- **On Error Page** - Error displays on a separate error page.

If the application contains a manually created tabular form (using `APEX_ITEM` calls in the SQL statement), select **On Error Page**. Otherwise, changes made by the user are discarded when the page is re-rendered. When errors are displayed on a separate error page, the user can use the browser's Back button to preserve the entered values.

3. Click **Apply Changes**.



See Also:

["Editing Application Process Attributes"](#)

Viewing the Application Processes History Report

The Application Process History report displays recent modifications made to application processes.

To access application processes History reports:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Logic, select **Application Processes**.
The Application Processes page appears.
3. Select the History tab at the top of the page.
4. You can customize the appearance the page using the Search bar at the top of the page. Available controls include:
 - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) to search for a process by name and click **Go**.
 - **Go button** - Executes a search or applies a filter.
 - **Actions menu** - Use the Actions menu to customize the report view.



See Also:

["About the Actions Menu"](#)

Understanding Application Computations

Create an application computation to set the value of a single page or application-level item. Application computations run at the same point across multiple pages in an application.



Tip:

To learn more about execution behavior in the case of a validation error, see ["What Happens When a Validation Fails?."](#)

- [About Application Computations](#)
- [About Application Computations that Execute On New Instance](#)
- [Creating an Application Computation](#)
- [Accessing the Application Computation History Report](#)

About Application Computations

Application Computations are units of logic that set the value of a single page or application-level item and are run at the same point across multiple pages in an application. Like page-level computation, application computations can be based on static values, item values, PL/SQL, or SQL.

A common use of an application item is to store the value of the last page viewed in the application. By storing the value in an item, you can add a back button and then redirect the user to the page number captured by the computation. This type of computation works well, for example, when you need to enable users to back out of an error page.

The following is an example of a computation that stores the last page visited. In this example, the computation:

- Stores the last application page visited to an item named `LAST_PAGE`
- Checks that the value of a `CURRENT_PAGE_ITEM` is of type PL/SQL Function Body with a Computation body of:

```
BEGIN
  :LAST_PAGE := nvl(:CURRENT_PAGE, :APP_PAGE_ID);
  :CURRENT_PAGE := :APP_PAGE_ID;
  RETURN :LAST_PAGE;
END;
```

About Application Computations that Execute On New Instance

Typically an application computation runs at the same point across multiple pages in an application. The exception is computations having a Computation Point of **On New Instance**. These types of computations are useful when you only need to retrieve information once within a user's session. For example, if the application items do not depend on the logged in user, you can initialize them using the Computation Point **On New Instance**. To look up information that depends on the user, you can also use **After Authentication**.

If you configure applications to share the same session by setting the same cookie name in the authentication scheme, **On New Instance** and **After Login**, the application processes fire whenever Oracle Application Express first processes a request for the application (that is, **On New Instance**), or processes a request of an authenticated user for the application (that is, **After Authentication**).

Creating an Application Computation

To create an application computation:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Logic, select **Application Computations**.
3. Click **Create**.
4. Edit the following attributes:
 - a. Item, Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - b. Item, Computation Item - Select the item this computation affects.
 - c. Frequency, Computation Point - Select a process point at which this computation should be performed. Selecting **After Submit** causes the computation to be performed only after the page is displayed and then submitted.
 - d. Computation, Computation Type - Select the manner in which this computation is performed.
 - e. Computation, Computation - Enter the computation logic that corresponds to the computation type. See field-level Help for examples.
 - f. Computation, Computation Error Message - Enter the error message that displays if the computation fails.
 - g. Authorization, Authorization Scheme (optional) - Select an authorization scheme which must evaluate to True in order for this computation to execute.
 - h. Condition, Condition Type (optional) - Select a condition type that must be met in order for this computation to execute.
 - i. Build Option (optional) - Select a build option for this component.

5. Click **Create Computation**.



See Also:

["Using Build Options to Control Configuration"](#)

Accessing the Application Computation History Report

Once you create an application computation, you can view the Application Computation History report. The Application Computation History report displays a history of recently changed application computations by date.

To access the Application Computation History report:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Logic, select **Application Computations**.
3. Select the **History** at the top of the page.
4. You can customize the appearance the page using the Search bar at the top of the page. Available controls include:
 - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
 - **Go button** - Executes a search or applies a filter.
 - **Actions menu** - Use the Actions menu to customize the report view.



See Also:

["About the Actions Menu"](#)

Using the Attribute Dictionary

Use the Manage Attribute Dictionary page to update values on the current page with those in the Attribute Dictionary, or to update the Attribute Dictionary with the values on the current page.

- [About the Attribute Dictionary](#)
- [Accessing the Manage Attribute Dictionary](#)

About the Attribute Dictionary

The Attribute Dictionary contains a set of attributes about a column that are used in creating forms and reports. The definitions are matched by column name and a particular definition can be shared among several columns by using synonyms.

Accessing the Manage Attribute Dictionary

To access the Manage Attribute Dictionary page:

1. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
2. Select an application.
3. Click **Utilities**.
4. Click **Utilities** and click **Attribute Dictionary**.

The page is divided into two sections: Page Items and Report Columns.

Use Page Item and Report Column definitions to update the Attribute Dictionary. You can also use the Attribute Dictionary to update Page Items and Report Columns.

Hidden objects (that is, those in hidden regions and button items) are not counted in the number of candidate Items and Report Columns as these are not used in the updates.

5. To update the current page or update the Attribute dictionary, select the appropriate page link.

The Attribute Dictionary appears.

Only Items and Report Columns that have different values for at least one attribute are shown. You choose which objects to insert or update and which attributes to include. The number of Items and Report Columns that can be used might be less than the total because hidden objects, objects in hidden regions, and button items are not included.

Tip:

You can access the Attribute Dictionary in Page Designer by clicking **Utilities** and then selecting **Attribute Dictionary**.

Managing Shared Components

Shared components can display or be applied on any page within an application. Developers can use the tools and wizards on the Shared Components page either at the application-level or on specific pages.

- [Accessing the Shared Components Page](#)
Access the Shared Components page by selecting an application and then clicking **Shared Components**.
- [Shared Components Page](#)
Use the tools and wizards on the Shared Components page either at the application-level or on specific pages.
- [Managing Application Settings](#)
Create Application Settings to define application-level configuration options.
- [Creating Lists of Values at the Application-Level](#)
Create a static or dynamic list of values (LOV) at the application-level which can be referenced by page items or report fields.
- [Managing Component Settings](#)
Use Component Settings to set application level values for built-in Application Express components and installed plug-ins.
- [Using Shortcuts](#)
Use shortcuts to avoid repetitive coding of HTML or PL/SQL functions.
- [Managing Email Templates](#)
Enable users to send emails from your application by creating email templates.
- [Managing Static Application Files](#)
Learn how to upload, view, download, and delete static application files (including images, CSS files, and other files which must be managed independently).
- [Managing Static Workspace Files](#)
Learn how to upload, view, download, and delete static workspace files (including images, CSS files, and other files which must be managed independently).
- [Managing Application-Level Items](#)
Application-level items do not display, but are used as global variables to the application. You can use an application item as a global variable.
- [Creating Lists](#)
Add a shared collection of links (or a list) to a database application. To add a list to a page, create a region and specify the region type as List. You control the appearance of a list through list templates.
- [Managing Report Output](#)
Learn how to print a report region by defining a report query and how to use a report layout to format a report region.

 **See Also:**

- ["App Builder Concepts"](#)
- ["Creating Database Applications"](#)
- ["Controlling Page Layout"](#)
- ["Adding Navigation"](#)

Accessing the Shared Components Page

Access the Shared Components page by selecting an application and then clicking **Shared Components**.

To access the Shared Components page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components** in the center of the page.

The Shared Components page appears.

4. To create a shared component, select the appropriate link.

 **Tip:**

Once you select an application, you can also access the Shared Components page by clicking the Shared Components icon at the top of the page. The Shared Components icon consists of a triangle above a circle and a square. This icon displays at the top of most App Builder pages including the Application home page, Page Designer, Supporting Objects, and Utilities.

Shared Components Page

Use the tools and wizards on the Shared Components page either at the application-level or on specific pages.

The Shared Components page is divided into regions.

- [Application Logic](#)
- [Security](#)
- [Other Components](#)
- [Navigation](#)
- [User Interface](#)
- [Files](#)

- [Data Sources](#)
- [Reports](#)
- [Globalization](#)
- [Tasks Region](#)
- [Workspace Objects Region](#)

Application Logic

The following table describes the links under **Application Logic** on the Shared Components page.

Table 17-1 Application Logic Links on the Shared Components Page

| Link | Description |
|-----------------------------------|---|
| Application Definition Attributes | Links to the Edit Application Definition page. Use this page to edit attributes for an application. Attributes are grouped into four categories: Definition, Security, Globalization, and User Interface. See Also: " Managing Application Attributes " |
| Application Items | Application-level items do not display, but are used as global variables to the application. Commonly, you set the value of a page-level item using an application or page computations. . See Also: " Managing Application-Level Items " |
| Application Processes | Use application processes to run PL/SQL logic: <ul style="list-style-type: none"> • At specific points for each page in an application • As defined by the conditions under which the process is set to execute • Upon the creation of a new session Note that On Demand processes execute only when called from a page-level On Demand process or when called using Ajax from the browser. See Also: " Understanding Application Processes " |
| Application Settings | Application Settings enable developers to define application level configuration options. See Also: " Managing Application Settings " |
| Application Computations | Use application-level computations to assign values to application and page-level items for each page displayed or upon the creation of a new application session. You can also create an application-level computation and execute it conditionally on multiple pages. See Also: " Understanding Application Computations " |
| Build Options | Use build options to conditionally display or process specific functionality within an application. You can use build options to control which features of an application are turned on for each application deployment. See Also: " Using Build Options to Control Configuration ". |

Security

The following table describes the links under **Security** on the Shared Components page.

Table 17-2 Security Links on the Shared Components Page

| Link | Description |
|----------------------------|--|
| Security Attributes | Use the Edit Security Attributes page to configure general security attributes for all pages within an application. See Also: " Configuring Security Attributes " |
| Authentication Schemes | Authentication is the process of establishing each user's identity before they can access your application. Authentication may require a user to enter a user name and password or may involve verification of a user's identity or use of a secure key. See Also: " Establishing User Identity Through Authentication " |
| Authorization Schemes | Authorization restricts user access to specific controls or components based on predefined user privileges. See Also: " Providing Security Through Authorization " |
| Session State Protection | Session State Protection is a built-in functionality that prevents hackers from tampering with the URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy. See Also: " Preventing URL Tampering " |
| Web Credentials | Web Credentials to connect to REST Enabled SQL or other external REST services. Oracle Application Express stores these securely stores and encrypts these credentials for use by Oracle Application Express components. Credentials cannot be retrieved back in clear text. See Also: " Managing Web Credentials " |
| Application Access Control | Manage application access control roles and user role assignments. See Also: " Managing Roles and User Assignments " |

Other Components

The following table describes the links under **Other Components** on the Shared Components page.

Table 17-3 Other Component Links on the Shared Components Page

| Link | Description |
|----------------|---|
| List of Values | A list of values (LOV) is a static or dynamic set of values used to display a popup list of values, select list, check box, or radio group. See Also: " Creating Lists of Values at the Application-Level " |

Table 17-3 (Cont.) Other Component Links on the Shared Components Page

| Link | Description |
|--------------------|---|
| Plug-ins | App Builder includes built-in item types, region types, dynamic actions, and processes. Use plug-ins to add new declarative types in to your application. See Also: " Implementing Plug-ins " |
| Component Settings | Use Component Settings to set application-level values for built-in Oracle Application Express components and installed plug-ins. See Also: " Managing Component Settings " |
| Shortcuts | Use shortcuts to avoid repetitive coding of HTML or PL/SQL functions. You can create a shortcut to define a page control such as a button, HTML text, a PL/SQL procedure, or HTML. Once you define a shortcut, it is stored in a central repository so you can reference it from various locations within your application. See Also: " Using Shortcuts " |
| Email Templates | Create templates to define the HTML format and Plain Text formats for the emails you wish to send from an application. See Also: " Managing Email Templates " |

Navigation

The following table describes the links under **Navigation** on the Shared Components page.

Table 17-4 Navigation Links on the Shared Components Page

| Link | Description |
|-----------------|--|
| Lists | A list is a shared collection of links. You control the appearance of a list through list templates. Each list element has a display condition that enables you to control when it displays. See Also: " Creating Lists " |
| Navigation Menu | A navigation menu is a list links that enables users to navigate the pages in an application. Navigation menus are only supported in applications using the <i>Universal Theme - 42</i> . See Also: " Managing Navigation Menus " |
| Tabs | Note: Tabs only appear if the associated application uses an older theme. In new themes, Tabs have been replaced with Navigation Menu. Tabs are an effective way to navigate users between pages in an application. You can create two types of tabs: standard tabs or parent tabs. A standard tab set is associated with a specific page and page number. A parent tab set functions as a container to hold a group of standard tabs. See Also: " Creating Tabs " |

Table 17-4 (Cont.) Navigation Links on the Shared Components Page

| Link | Description |
|---------------------|---|
| Breadcrumbs | Breadcrumbs provide users with hierarchical navigation. A breadcrumb is a hierarchical list of links that display using templates. You can display a breadcrumb as a list of links or as a breadcrumb path. See Also: " Creating Breadcrumbs " |
| Navigation Bar List | Navigation bar lists offer users a simple navigation path for moving between pages in an application. The location of a navigation bar depends upon the associated page template. A list entry can be an image, an image with text beneath it, or text. See Also: " Creating Classic Navigation Bar " |

User Interface

The following table describes the links under **User interface** on the Shared Components page.

Table 17-5 User interface Links on the Shared Components Page

| Link | Description |
|---------------------------|--|
| User Interface Attributes | Use User Interface page to specify user interface options for an application. See Also: " Managing the Application User Interface " |
| Themes | A theme is a named collection of templates that defines the application user interface. See Also: " Using Themes " |
| Templates | Templates control the look and feel of specific constructs within your application, such as pages, regions, items, and menus. See Also: " Creating Custom Themes " |

Files

The following table describes the links under **Files** on the Shared Components page.

Table 17-6 Files Links on the Shared Components Page

| Link | Description |
|--------------------------|--|
| Static Application Files | Use this link to upload, edit, and delete static files associated with the current application, including style sheets (CSS), images and JavaScript files. See Also: " Managing Static Application Files " |
| Static Workspace Files | Use this link to upload, edit, and delete static files associated with the current application, including style sheets (CSS), images and JavaScript files. See Also: " Managing Static Workspace Files " |

Data Sources

The following table describes the links under **Data Sources** on the Shared Components page.

Table 17-7 Links Under Data Sources on the Shared Components Page

| Link | Description |
|--------------------------------------|---|
| Data Load Definitions | Use Data Load Tables to define tables for use in the Data Loading create page wizard. A Data Load Table is an existing table in your schema that has been selected for use in the data loading process to upload data. See Also: "Importing a Plug-in from the Plug-in Page" |
| REST Enabled SQL | Use REST Enabled SQL to execute SQL or PL/SQL defined in Application Express components on a remote Oracle database. See Also: "Managing REST Enabled SQL References" |
| Web Source Modules | Web Source Modules act as a reference to one or multiple external web services. A module can contain one or many Web Source Operations which are the references to a concrete external web service. Configurations at the module level are shared across all operations of this module. See Also: "Managing Web Source Modules" |
| Legacy Web Service References (SOAP) | The Application Express engine can use Web Service References to access a Web service across the network. The Web service performs an action and then sends back a response. See Also: "Managing Legacy Web Services" |

Reports

The following describes the links under User interface on the Shared Components page.

Table 17-8 Data References Links on the Shared Components Page

| Link | Description |
|----------------|---|
| Report Queries | Use the Report Queries link to view a report of stored queries within the current application. See Also: "Printing a Report Region by Defining a Report Query" |
| Report Layouts | Use Report Layouts with a report or shared query to render data in a printer-friendly format, such as Adobe Portable Document Format (PDF), Microsoft Word Rich Text Format (RTF), or Microsoft Excel (XLS) format. See Also: "Formatting a Report Region or Report Query Using Report Layouts" |

Globalization

The following table describes the links under **Globalization** on the Shared Components page.

Table 17-9 Globalization Links on the Shared Components Page

| Link | Description |
|--------------------------|---|
| Globalization Attributes | You can develop applications that can run concurrently in different languages. Click this link to specify globalization options such as the Application Primary Language and Application Language Derived From attributes. See Also: " Configuring Globalization Attributes " and " Understanding Application Translation and Globalization Support " |
| Text Messages | Text messages are named text strings that can be called from the PL/SQL code you write. This PL/SQL can be anonymous blocks within page processes and page regions, or in packages and procedures. See Also: " Translating Messages " |
| Translate Application | You can develop applications in Oracle Application Express that can run concurrently in different languages. A single Oracle database and Oracle Application Express instance can support an application in multiple languages. Translating an application involves multiple steps. See Also: " Understanding Application Translation and Globalization Support " and " Understanding the Translation Process " |

Tasks Region

The following table describes the Tasks region on the right side of the Shared Components page.

Table 17-10 Tasks

| Link | Description |
|-------------------------------|--|
| Export Application Components | Links to the Component Export page. Use this page to identify the components of an application to be exported either for backup purposes or to load the components on another instance. See Also: " Exporting Application Components " |
| Manage Supporting Objects | Links to the Supporting Objects page. Use this page to create a custom application. See Also: " How to Create a Custom Application " |
| Edit Application Comments | Links to the Application Comments page. Use this page to enter comments specific to the currently selected application. See Also: " Adding Database Application Comments " |
| Developer Comments | Links to Developer Comments page. See Also: " Adding Developer Comments " |

Workspace Objects Region

The following table describes the Workspace Objects region on the right side of the Shared Components page.

Table 17-11 Workspace Objects

| Link | Description |
|------------------------|---|
| Static Workspace Files | Static workspace files are available to all applications for a given workspace. Use #WORKSPACE_IMAGES# in your application to reference a file. See Also: "Exporting Application Components" |
| Web Credentials | Use Web Credentials to connect to REST Enabled SQL or other external REST services. Oracle Application Express stores these securely and encrypts these credentials for use by Oracle Application Express components. Credentials cannot be retrieved back in clear text. See Also: "Managing Web Credentials" |
| REST Enabled SQL | Use REST Enabled SQL to execute SQL or PL/SQL defined in Application Express components on a remote Oracle database. See Also: "Managing REST Enabled SQL References" |
| Remote Servers | Links to the Remote Servers page. Remote Servers can be shared among multiple Web Sources. Remote Servers are stored at the Workspace-level and therefore visible in all applications. When an application is being exported, the used Remote Servers are being added to the export file. See Also: "Managing Remote Servers" |

Managing Application Settings

Create Application Settings to define application-level configuration options.

 **Tip:**

You also use the `APEX_APP_SETTING` API to access application settings.

- [Creating an Application Setting](#)
Create an Application Settings. Application Settings enable developers to define application level configuration options.
- [Editing or Deleting Application Setting](#)
Edit or delete an existing setting on the Application Settings page.

Creating an Application Setting

Create an Application Settings. Application Settings enable developers to define application level configuration options.

To create an Application Setting:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Settings**.
3. Click **Create Setting**.
4. On Application Setting:
 - a. Name - Enter a descriptive name for this application setting.
 - b. Value - Enter the default value for this application setting.
 - c. Value Required - Select whether this Application Setting requires a value. Select **Yes** or **No**.
 - d. Valid Values - Enter comma separated valid values. If defined, Application Express only accepts valid values when the settings change.
 - e. On Upgrade Keep Value - Select whether the application setting value in the deployed application should be kept, or overwritten with the specified value when the application is upgraded. Selecting **Yes** to keep the deployed value is useful if the application supports application setting configuration (using the `APEX_APP_SETTING.SET_VALUE` API) to change the application setting value in the deployed application, and where you want to respect those values upon upgrade.
 - f. Comments - Enter comments that describe this setting.
5. Click **Create Application Setting**.

The new Application Setting displays on the Application Settings page.

Editing or Deleting Application Setting

Edit or delete an existing setting on the Application Settings page.

To edit or delete an Application Setting:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Settings**.
The Application Settings page appears.

3. To edit an existing setting:
 - a. Select the setting.
The Application Setting dialog appears.
 - b. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help.
 - c. Click **Apply Changes**.
4. To delete an existing setting:
 - a. Select the setting.
The Application Setting dialog appears.
 - b. Click **Delete**.

Creating Lists of Values at the Application-Level

Create a static or dynamic list of values (LOV) at the application-level which can be referenced by page items or report fields.

Tip:

To learn about defining a column as a list of values, see the “Defining a Column as a List of Values” discussion under the report type in [“Developing Reports”](#)

- [What is a List of Values?](#)
A list of values (LOV) is a static or dynamic definition used to display a specific type page item.
- [Creating a Named LOV at the Application-Level](#)
You define named (or shared) LOVs at the application-level by running the Create LOV Wizard and adding them to the List of Values repository.
- [About Static LOVs](#)
Static LOVs are based on a static list of display values and return values you specify when you run the Create LOV Wizard.
- [Editing an Existing LOV](#)
To edit an existing LOV, select the LOV on the Lists of Values page.
- [About Referencing Session State Within an LOV](#)
You can reference session state in an LOV by using bind variables.
- [Accessing LOV Reports](#)
Access LOV reports by selecting a report on the List of Values page.

 **See Also:**

- ["Creating Page-Level Items"](#)
- ["Managing Shared Components"](#)

What is a List of Values?

A list of values (LOV) is a static or dynamic definition used to display a specific type page item.

A LOV can be *static*, meaning it is based on values the user enters or *dynamic*, meaning it is based on a SQL query. A LOV can be referenced by page items such as popup lists of values, a select list, a check box, a radio group, or multiple select lists.

By creating a list of values at the application-level, you are creating a shared component. Creating a LOV as a shared component has several advantages:

- You can add it to any page within an application.
- All LOV definitions are stored in one location, making them easy to locate and update.

Creating a Named LOV at the Application-Level

You define named (or shared) LOVs at the application-level by running the Create LOV Wizard and adding them to the List of Values repository.

All LOVs can be defined as **static** or **dynamic**. Static lists are based on predefined pairs of display values and return values. Dynamic lists are based on a SQL query you write that selects values from a table.

To create a named LOV:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Other Components, select **List of Values**.

The Lists of Values page appears.

3. To create a LOV, click **Create**.
4. Follow the on-screen instructions.

New named LOVs are added to the List of Values repository. To add the LOV to a page, you must edit the appropriate item or column and select the named LOV.

About Static LOVs

Static LOVs are based on a static list of display values and return values you specify when you run the Create LOV Wizard.

To create a static LOV, run the Create LOV Wizard and select the LOV type **Static**. Oracle Application Express stores the display values, return values, and sort sequence you specify in the List of Values repository. Once you add a static LOV to the repository, you can create an item and display it as a check box, radio group, select list, or popup list based on this definition.

Editing an Existing LOV

To edit an existing LOV, select the LOV on the Lists of Values page.

- [Editing a LOV](#)
- [Editing LOVs in Bulk](#)

Editing a LOV

To edit a LOV:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Other Components, select **List of Values**.

The Lists of Values page appears.

3. Select an LOV.

The Edit page appears.

4. Edit the attributes.



Tip:

To learn more about an attribute, see field-level Help.

5. Click **Apply Changes**.



See Also:

["Viewing Field-Level Help"](#)

Editing LOVs in Bulk

You can edit the display values of all static LOVs by clicking the Grid Edit button on the Edit List of Values page.

To perform a bulk edit of static LOVs:

1. Navigate to the Shared Components page:

- a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, select **List of Values**.
By default, LOVs display as icons.
 3. Click the **Grid Edit** tab.
 4. Edit the display text and click **Apply Changes**.

About Referencing Session State Within an LOV

You can reference session state in an LOV by using bind variables.

Keep in mind that referencing session state makes an LOV a bit less reusable, but is still a recommended development practice. In the following example, this LOV only works if the item called *my_deptno* contains a valid department number.

```
SELECT ename, empno FROM emp WHERE deptno = :P1_DEPTNO
```

Accessing LOV Reports

Access LOV reports by selecting a report on the List of Values page.

To access LOV reports:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, select **List of Values**.
By default, LOVs display as icons.
3. Select one of the following tabs at the top of the page:
 - **List of Values** - Displays all LOVs that are defined locally with the definition of an item. LOVs that are usable by multiple pages can be converted into shared lists of values. Converting locally defined lists of values into a shared list of values promotes reuse.
 - **Edit Null Text** - Edit null text for all LOVs in your application.
 - **Locally Defined** - View all lists of values that are defined locally with the definition of the item. Lists of values that are usable by multiple pages can be converted into shared lists of values. Converting locally defined lists of values into a shared list of values promotes reuse..
 - **Grid Edit** - Find and edit static list of value entries over all static lists of values. Use this page to standardize display text.
 - **Conditional Entries** - Displays all LOVs that display conditionally.
 - **Subscription** - Displays all subscribed LOVs in your application.

- **Utilization** - Displays where LOVs used in the current application.
 - **History** - Displays displays recent modifications made to lists of values in in the current application.
4. Follow the on-screen instructions.

Managing Component Settings

Use Component Settings to set application level values for built-in Application Express components and installed plug-ins.

Tip:

The Component Settings in your environment will vary based on the currently selected application.

- [Configuring Component Settings](#)
Developers can configure Component Settings to set application level values for built-in Application Express components and installed plug-ins.
- [Integrating with Google Calendar](#)
Developers can enable calendars within an application to integrate with Google calendar data by configuring the Component Settings, Calendar.
- [Configuring Color Picker Display](#)
Developers can control the Color Picker display style by configuring the Component Setting, Color Picker.
- [Configuring Interactive Report Action Menu Structure](#)
Developers can configure Component Settings, Interactive Report to use the new structure or keep the legacy structure from earlier releases.
- [Dynamically Adding Sliders to Regions](#)
Developers can dynamically add a slider to the region if the displayed region name exceeds the width of the region.
- [Configuring Switch Defaults](#)
The Switch item type displays as a flip toggle switch. Developers can configure default switch behavior by configuring the Component Settings, Switch.
- [Configuring Rich Text Editor Responsiveness](#)
Developers can configure whether the Rich Text Editor automatically adjusts to the width of the container region by configuring Component Settings, Rich Text Editor.
- [Viewing the Component Settings History Report](#)
You can view a report of modifications made to Oracle Application Express components and installed plug-ins on the Component Settings History page.

Configuring Component Settings

Developers can configure Component Settings to set application level values for built-in Application Express components and installed plug-ins.

To configure Component Settings:

1. On the Workspace home page, click **App Builder**.

2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Other Components, click **Component Settings**.
5. Click the component name.
6. In the Component Settings configure the appropriate settings and click **Apply Changes**.

 **Tip:**

To learn more about a specific setting, see item Help.

Integrating with Google Calendar

Developers can enable calendars within an application to integrate with Google calendar data by configuring the Component Settings, Calendar.

To configure the Component Settings, Calendar:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Other Components, click **Component Settings**.
5. Click the component name, **Calendar**.
6. In the Component Settings dialog:
 - a. Google Calendar API Key - Enter the Google Calendar API key to enable calendars within this application to integrate with Google calendar data.
 - b. Click **Apply Changes**.

Configuring Color Picker Display

Developers can control the Color Picker display style by configuring the Component Setting, Color Picker.

To configure the Component Settings, Color Picker:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Other Components, click **Component Settings**.
5. Click the component name, **Color Picker**.
6. In the Component Settings dialog:

- a. Display Style - Options include:
 - **Modern** - Renders the color preview as a small circle inside the text field.
 - **No Preview** - Do not render a preview of the chosen color.
- b. Click **Apply Changes**.

Configuring Interactive Report Action Menu Structure

Developers can configure Component Settings, Interactive Report to use the new structure or keep the legacy structure from earlier releases.

The structure of the interactive report Actions menu has changed to be consistent with interactive grids.

To configure the Component Settings, Interactive Report:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Other Components, click **Component Settings**.
5. Click the component name, **Interactive Report**.
6. In the Component Settings dialog:
 - a. Actions Menu Structure - Specify how the Actions menu displays in interactive reports. Options include:
 - Interactive Grid - Select **Interactive Grid** to have the interactive report Actions menu match interactive grid.
 - Legacy - Select **Legacy** to have the interactive report Actions menu use the legacy structure.
 - b. Click **Apply Changes**.

Dynamically Adding Sliders to Regions

Developers can dynamically add a slider to the region if the displayed region name exceeds the width of the region.

To configure Component Settings, Region Display Selector:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Other Components, click **Component Settings**.
5. Click the component name, **Region Display Selector**.
6. In the Component Settings dialog:
 - a. Include Slider - Select **Yes** to dynamically add a slider to the region if the displayed region name exceeds the width of the region.

- b. Click **Apply Changes**.

Configuring Switch Defaults

The Switch item type displays as a flip toggle switch. Developers can configure default switch behavior by configuring the Component Settings, Switch.

To configure the Component Settings, Switch:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Other Components, click **Component Settings**.
5. Under Component Settings, click **Switch**.
6. In the Component Settings dialog:
 - a. On Value - Enter the value stored in the page item if the end user picks the **On** option.
 - b. On Label - Enter the text displayed for the **On** option. If no text is entered, the system default label is used.
 - c. Off Value - Enter the value stored in the page item if the user picks the **Off** option.

 **Note:**

This value is used if the page item is populated with a value which does not match either the **On Value** or **Off Value**.

- d. Off Label - Enter the text displayed for the **Off** option. If no text is entered, the system default label is used.
- e. Click **Apply Changes**.

Configuring Rich Text Editor Responsiveness

Developers can configure whether the Rich Text Editor automatically adjusts to the width of the container region by configuring Component Settings, Rich Text Editor.

For new apps, this setting is enabled by default. For older apps (pre-18.1), this is disabled by default.

This setting requires a responsive application theme (such as Universal Theme).

To configure Component Settings, Rich Text Editor:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.

4. Under Other Components, click **Component Settings**.
5. Under Component Settings, click **Rich Text Editor**.
6. For Responsive in the Component Settings dialog:
 - **Yes** - The Rich Text Editor is responsive and automatically adjusts its width to the container region.
 - **No** - The Rich Text Editor is not responsive. Choose this option for older apps that use a non-responsive legacy theme.
7. Click **Apply Changes**.

Viewing the Component Settings History Report

You can view a report of modifications made to Oracle Application Express components and installed plug-ins on the Component Settings History page.

To view the Component Settings History report:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Other Components, click **Component Settings**.
3. Click the **History** tab.

The History page displays as an interactive report. To customize the report, use the Search bar at the top of the page.

Using Shortcuts

Use shortcuts to avoid repetitive coding of HTML or PL/SQL functions.

- [What is a Shortcut?](#)
Use a shortcut to define frequently used code once and then reference it in many places thus reducing code redundancy. For example, you can create a shortcut to define a page control such as a button, HTML text, a PL/SQL procedure, or HTML.
- [Supported Shortcut Types](#)
Oracle Application Express supports six types of shortcuts.
- [Defining a Shortcut from Scratch](#)
Define a shortcut on the Shortcuts page.
- [Copying an Existing Shortcuts](#)
Copy existing shortcuts from the Shortcut repository.
- [Editing Existing Shortcuts](#)
Once you create a shortcut, you can alter it by editing attributes on the Edit Shortcut page.
- [Accessing Shortcut Reports](#)
View the shortcut Subscription and History reports.

What is a Shortcut?

Use a shortcut to define frequently used code once and then reference it in many places thus reducing code redundancy. For example, you can create a shortcut to define a page control such as a button, HTML text, a PL/SQL procedure, or HTML.

You can use a shortcut within the following locations:

- The Region Source attribute of regions defined as HTML Text (with shortcuts).
- Region Header and Footer Text attribute.
- Item Label attributes, Pre Element Text, Post Element Text, and Default Value attribute.
- Region Templates attributes.

Referencing a Shortcut

Once you define a shortcut, you can invoke it using specific syntax unique to the location in which the shortcut is used.

You reference shortcuts using the following syntax:

```
"MY_SHORTCUT"
```

The shortcut name must be capitalized and enclosed in quotation marks.



See Also:

- ["About Regions"](#)
- ["Specifying a Region Header and Footer"](#)
- ["Creating Custom Themes"](#)

Supported Shortcut Types

Oracle Application Express supports six types of shortcuts.

When you create a shortcut you select a shortcut type. Supported shortcut types include:

- PL/SQL Function Body
- HTML Text
- HTML Text with Escaped Special Characters
- Image
- Text with JavaScript Escaped Single Quotes
- Message
- Message with JavaScript Escaped Special Quotes

Text with JavaScript Escaped Single Quotes

Use this type of shortcut to reference a shortcut inside of a JavaScript literal string. This shortcut defines a text string. When the shortcut is referenced, it escapes the single quotation marks required for JavaScript.

Message

Use this type of shortcut to reference a translatable message at runtime. Since this shortcut does not have a shortcut body, the name of the shortcut must match the corresponding message name. At runtime, the name of the shortcut expands to the text of the translatable message for the current language.

Message with JavaScript Escaped Single Quotes

Use this type of shortcut to reference a shortcut inside of JavaScript literal string and reference a translatable message at runtime.



See Also:

["Understanding Application Translation and Globalization Support"](#)

Defining a Shortcut from Scratch

Define a shortcut on the Shortcuts page.

To define a shortcut:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, select **Shortcuts**.
3. Click **Create**.
4. For Create Shortcut, select **From Scratch** and click **Next**.
5. On Shortcut Attributes:
 - a. Name - Enter the name by which the shortcut will be referenced.
 - b. Type - Select a shortcut type. Options include:
 - PL/SQL Function Body
 - HTML Text
 - HTML Text with Escaped Special Characters
 - Image
 - Text with JavaScript Escaped Single Quotes

- Message
 - Message with JavaScript Escaped Special Quotes
 - c. Shortcut - Enter the text of the shortcut. The shortcut must be less than 30,000 characters in length.
 - d. Error Text - Enter the text to display if an error should occur during the processing of a shortcut.
 - e. Build Option - Select a build option for this component.
6. Click **Create**.

New shortcuts are added to the Shortcuts page (also referred to as the Shortcut repository).

Copying an Existing Shortcuts

Copy existing shortcuts from the Shortcut repository.

To copy an existing shortcut:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.The Shared Components page appears.
2. Under Other Components, select **Shortcuts**.
3. Click **Create**.
4. For Create Shortcut, select **As a Copy of an Existing Shortcut** and click **Next**.
5. On Copy From Application, Select the application you want to copy shortcuts from and click **Next**.
6. On New Shortcuts: Click **Copy Shortcut**.
 - a. To Name - Enter the new name of the shortcut to be copied.
 - b. Copy - Select **Yes**, **No**, or **Copy and Subscribe**.
 - c. Click **Copy Shortcut**.

New shortcuts are added to the Shortcuts page (also referred to as the Shortcut repository).

Editing Existing Shortcuts

Once you create a shortcut, you can alter it by editing attributes on the Edit Shortcut page.

To edit an existing shortcut:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.

- c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
 2. Under Other Components, select **Shortcuts**.
 3. Select a shortcut.
The Edit Shortcut page appears.
 4. Edit the attributes.
To learn more about an attribute, see field-level Help.
 5. Click **Apply Changes**.



See Also:

["Viewing Field-Level Help"](#)

Accessing Shortcut Reports

View the shortcut Subscription and History reports.

To access shortcut reports:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, select **Shortcuts**.
3. Click one of the following tabs:
 - **Subscription** - Displays the Subscribed Shortcuts page which displays all subscribed shortcuts in your application.
 - **History** - Display the Shortcut History page which displays a history of recently changed shortcuts by date.



Note:

The Subscription and History tabs only appear after you create a shortcut.

Managing Email Templates

Enable users to send emails from your application by creating email templates.

 **Note:**

Before you can send email from an application, your Instance administrator must log in to Oracle Application Express Administration Services, navigate to the Instance Settings page and configure Email attributes.

- [Defining an Email Template](#)
Define email templates for your application. Email templates include both an HTML Format and a Plain Text Format.
- [Viewing Sample Email Templates](#)
View the sample email templates: Load Order Details, Load Event Reminder, or Load Scheduled Outage.
- [About Including Dynamic Text](#)
Provide dynamic text in your email templates using existing substitution modifiers.
- [Editing Email Templates](#)
Edit or delete existing email templates.

 **See Also:**

- "Configuring Email" in *Oracle Application Express Administration Guide*
- "APEX_MAIL" in *Oracle Application Express API Reference*

Defining an Email Template

Define email templates for your application. Email templates include both an HTML Format and a Plain Text Format.

To define an email template:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, select **Email Templates**.
3. On the Email Templates page, click **Create Email Template**.
The Details page appears.

 **Tip:**

To get started, review the sample email templates available under **Sample Templates** on the right side of the page. To load a sample email template, simply select it. See "[Viewing Sample Email Templates](#)".

4. Next, define your template.

 **Tip:**

For **Email Subject**, **HTML Format** or **Plain Text Format**, support substitution strings using the format `#STRING_NAME#` . You can pass in values for these strings using the `APEX_MAIL` API.

5. Under Identification:

- a. Template Name - Enter a descriptive name for this template .
- b. Static Identifier - Static string identifier used to refer to the template when calling the `APEX_MAIL` APIs.
- c. Email Subject - Enter the text to display for the email subject.

6. Under HTML Format:

- a. Define the Header, Body, and Footer. Body supports basic HTML markup.
- b. Under Advanced, optionally click **Load Default HTML**.

The default HTML appears in the HTML Template. If needed, edit the default HTML Template.

7. Under Plain Text Format, enter the appropriate template defaults.

8. Under Comments, optionally enter comments that describe this template.

9. Click **Create Email Template**.

Next, create a button and process to call the `APEX_MAIL` API.

10. Add a "Send Mail" button:

- a. Return to the application from which you want to send email.
- b. Navigate to the appropriate page or create a new page.
- c. Add a button to send the email.
- d. Create a PL/SQL process which calls the `APEX_MAIL` API.

Consider the following example:

```
begin
  apex_mail.send (
    p_to           => 'steven.king@example.com',
    p_template_static_id => 'ORDER_CONFIRMATION',
    p_placeholders => q'~
    {
      "CUSTOMER_NAME": "Steven King",
      "ORDER_NUMBER": 1234,
      "ORDER_DATE": "02-Feb-2018",
```

```
        "SHIP_TO": "Steven King",
        "SHIPPING_ADDRESS_LINE_1": "2004 Charade Rd",
        "SHIPPING_ADDRESS_LINE_2": "Seattle, Washinton",
        "ITEMS_ORDERED": 3,
        "ORDER_TOTAL": "$ 1,200.99",
        "ORDER_URL": "http://domain/apex/f?p=&APP_ID"
    }~' );
    apex_mail.push_queue;
end;
```

- e. Run the page and click the **Send Mail** button.

See Also:

- ["Managing Pages in a Database Application"](#)
- ["Managing Buttons"](#)
- ["Understanding Page Processes"](#)
- "APEX_MAIL" in *Oracle Application Express API Reference*

Viewing Sample Email Templates

View the sample email templates: Load Order Details, Load Event Reminder, or Load Scheduled Outage.

To view sample email template:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, select **Email Templates**.
3. On the Email Templates page, click **Create Email Template**.
The Details page appears.
4. On the right side of the page, find the **Sample Templates** region.
The Sample Templates region contains three templates:
 - Load Order Details
 - Load Event Reminder
 - Load Scheduled Outage
5. Select a template. For example, select **Load Order Details**.
The template appears.

6. Expand the **Sample API Usage** region at the bottom of the page to see an example. The following is the **Load Order Details** API example.

```

begin
  apex_mail.send (
    p_to                => email_address_of_user,
    p_template_static_id => '',
    p_placeholders      => '{'
      'CUSTOMER_NAME': '      ||
apex_json.stringify( some_value )
      ', "ITEMS_ORDERED": '      ||
apex_json.stringify( some_value )
      ', "MY_APPLICATION_LINK": '      ||
apex_json.stringify( some_value )
      ', "ORDER_DATE": '      ||
apex_json.stringify( some_value )
      ', "ORDER_NUMBER": '      ||
apex_json.stringify( some_value )
      ', "ORDER_TOTAL": '      ||
apex_json.stringify( some_value )
      ', "ORDER_URL": '      ||
apex_json.stringify( some_value )
      ', "SHIPPING_ADDRESS_LINE_1": '      ||
apex_json.stringify( some_value )
      ', "SHIPPING_ADDRESS_LINE_2": '      ||
apex_json.stringify( some_value )
      ', "SHIP_TO": '      ||
apex_json.stringify( some_value )
      '}' );
end;
```

About Including Dynamic Text

Provide dynamic text in your email templates using existing substitution modifiers.

For security reason, the values substituted for the placeholders (#NAME#) in the email templates are automatically escaped based on the context. In other words, HTML formatted templates include HTML escaping and Plain Text templates do not. If this is not the desired behavior for your environment, you can provide dynamic text which contains HTML tags using the existing substitution modifiers:

- #NAME!RAW# - Text is substituted as is.
- #NAME!STRIPHTML# - All HTML tags are removed.

Extending the Email Templates

For example, you can extend the **HTML Format - Body** template with:

```

<tr>
  left">Additional Information</th>
  <td>#ADDITIONAL_INFO!RAW#</td>
</tr>
```

Similarly, you can extend the **Plain Text Format** template with:

```
Additional Info: #ADDITIONAL_INFO!STRIPHTML#
```

Sample PL/SQL

The PL/SQL code would look similar to this:

```
begin
  apex_mail.send (
    p_to                => 'steven.king@example.com',
    p_template_static_id => 'ORDER_CONFIRMATION',
    p_placeholders      => q'~
    {
      "CUSTOMER_NAME": "Steven King",
      "ORDER_NUMBER": 1234,
      "ORDER_DATE": "02-Feb-2018",
      "SHIP_TO": "Steven King",
      "SHIPPING_ADDRESS_LINE_1": "2004 Charade Rd",
      "SHIPPING_ADDRESS_LINE_2": "Seattle, Washinton",
      "ITEMS_ORDERED": 3,
      "ORDER_TOTAL": "$ 1,200.99",
      "ORDER_URL": "http://domain/apex/f?p=&APP_ID"
      "ADDITIONAL_INFO": "We plan to deliver your ordered items
<strong>tomorrow morning between 08:00 and 09:00</strong>
    }~' );
    apex_mail.push_queue;
end;
```

Editing Email Templates

Edit or delete existing email templates.

To edit or delete an email template:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, select **Email Templates**.
The Email Templates page appears.
3. To edit an email template:
 - a. Click the email template name.
 - b. Edit the appropriate attributes.
 - c. Click **Apply Changes**.
4. To delete an email template:
 - a. Click the email template name.

- b. Click **Delete**.

Managing Static Application Files

Learn how to upload, view, download, and delete static application files (including images, CSS files, and other files which must be managed independently).

- [Specifying the Location of Static Application Files](#)
- [Uploading Static Application Files](#)
- [Viewing the Static Application File Report](#)
- [Referencing Static Application Files](#)
- [Downloading Static Application Files](#)
- [Deleting Static Application Files](#)

See Also:

- ["Managing Static Workspace Files"](#)
- ["About Managing Images"](#)

Specifying the Location of Static Application Files

Application files are stored with your application definition in the Oracle database. For performance reasons, you can also store your application files on your Web server.

To verify the Static File Prefix attribute for an application:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under User Interface, select **User Interface Attributes**.
3. Click the **User Interface** tab.
4. In Static File Prefix, enter the virtual path the Web server uses to point to static application files. Leave this attribute blank to reference files stored with your application definition in the database. For performance reasons you can also store your application files on your Web server and reference them using a valid URL.
Example:

```
/myFiles/  
http://contentDeveliveyNetwork.com/myFiles/
```

5. Click **Apply Changes**.

Uploading Static Application Files

To upload a static application file:


1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Application Files**.
The Static Application Files page appears.
3. To upload a file, click **Upload File**.
4. On Upload:
 - a. Directory - Enter the name of the directory where the file should be stored. If no directory is specified, the file is stored in the root directory.
 - b. File - Enter the location of the file to be added to the application.
 - c. File Character Set - Indicate the character set encoding for the file to be uploaded. This attribute only applies to text files.
 - d. Unzip File - If **Yes**, the uploaded Zip file is automatically unzipped and stored in the specified directory. Note that the actual Zip file is not stored. This attribute is ignored if the uploaded file is not a Zip file.
5. Click **Upload** or **Upload and Upload Another**.

Viewing the Static Application File Report

To view details about an uploaded static application file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Application Files**.
The Static Application Files page appears.
3. If not already selected, click **View Report** icon.
A report appears. To customize the report, use the Search bar at the top of the page.
The Static Application Files report displays the filename, MIME type, file size, a download link, and substitution string syntax.
4. To view additional details, click the **Edit** icon. If the file is an image, the image displays.
5. To delete the file, click **Delete**.


6. To exit, click **Cancel**.

 **See Also:**
["About the Search Bar"](#)

Referencing Static Application Files

You can reference a static application file in your application using `#APP_IMAGES#` substitution string.

Static application files are only available to the currently selected application. To view sample syntax for uploaded images, see the Reference column on the Static Application Files report.

 **See Also:**

- ["Viewing the Static Application File Report"](#)
- ["About Using Substitution Strings"](#)
- ["APP_IMAGES"](#)
- ["Defining an Application Logo"](#)

Downloading Static Application Files

Learn how to download an uploaded static application file or download all static application files in a Zip.

- [Downloading a Static Application File](#)
- [Downloading All Static Application Files in a Zip](#)

Downloading a Static Application File

To download an uploaded static application file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Files, select **Static Application Files**.

The Static Application Files page appears.

3. If not already selected, click **View Report** icon.

A report appears.

4. Locate the file to be downloaded and click the **Download** link.

Downloading All Static Application Files in a Zip

To download all static application files in a Zip:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Files, select **Static Application Files**.

The Static Application Files page appears.

3. Click **Download as Zip**.

Deleting Static Application Files

Learn how to delete uploaded static application files.

- [Deleting a Static Application File](#)
- [Deleting All Static Application Files](#)

Deleting a Static Application File

To delete an uploaded static application file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Files, select **Static Application Files**.

The Static Application Files page appears.

3. If not already selected, click the **View Report** icon.

A report appears. To customize the report, use the Search bar at the top of the page.

4. Select the file to be deleted.
5. On the Edit page, click **Delete**.



See Also:

["About the Search Bar"](#)

Deleting All Static Application Files

To delete all uploaded static application files:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Application Files**.
The Static Application Files page appears.
3. Click **Delete All Files**.

Managing Static Workspace Files

Learn how to upload, view, download, and delete static workspace files (including images, CSS files, and other files which must be managed independently).

- [Uploading Static Workspace Files](#)
- [Viewing Static Workspace File Report](#)
- [About Referencing Static Workspace Files](#)
- [Downloading Static Workspace Files](#)
- [Deleting Static Workspace Files](#)

See Also:

- ["Managing Static Application Files"](#)
- ["About Managing Images"](#)

Uploading Static Workspace Files

To upload a static workspace file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Workspace Files**.
The Static Workspace Files page appears.

3. To upload a file, click **Upload File**.
4. On Upload:
 - a. Directory - Enter the name of the directory where the file should be stored. If no directory is specified, the file is stored in the root directory.
 - b. File - Enter the location of the file to be added to the application.
 - c. File Character Set - Indicate the character set encoding for the file to be uploaded. This attribute only applies to text files.
 - d. Unzip File - If **Yes**, the uploaded Zip file is automatically unzipped and stored in the specified directory. Note that the actual Zip file is not stored. This attribute is ignored if the uploaded file is not a Zip file
5. Click **Upload** or **Upload and Upload Another**.

Viewing Static Workspace File Report

To view the Static Workspace File report:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.The Shared Components page appears.
2. Under Files, select **Static Workspace Files**.

The Static Workspace Files page appears. To customize the report, use the Search bar at the top of the page.
3. If not already selected, click **View Report** icon.

The Static Application Files report displays the filename, MIME type, file size, a download link, and substitution string syntax.
4. To view additional details, select the file. If the file is an image, the image displays.
5. To delete the file, click **Delete**.
6. To exit, click **Cancel**.



See Also:

["About the Search Bar"](#)

About Referencing Static Workspace Files

You can reference static workspace files in any application in the current workspace. To reference a static workspace file in your application, use the `#WORKSPACE_IMAGES#` substitution string. To view sample syntax for uploaded images, see the Reference column on the Static Workspace Files report.

 **See Also:**

- ["Viewing Static Workspace File Report"](#)
- ["Using Built-in Substitution Strings"](#)
- ["WORKSPACE_IMAGES"](#)
- ["Defining an Application Logo"](#)

Downloading Static Workspace Files

Learn how to download static workspace files.

- [Downloading a Static Workspace File](#)
- [Downloading All Static Workspace Files in a Zip](#)

Downloading a Static Workspace File

To download an uploaded static workspace file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Workspace Files**.
The Static Workspace Files page appears.
3. If not already selected, click **View Report** icon.
A report appears.
4. Locate the file to be downloaded and click the **Download** link.

Downloading All Static Workspace Files in a Zip

To download all static workspace files in a Zip:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Workspace Files**.
The Static Application Files page appears.
3. Click **Download as Zip**.

Deleting Static Workspace Files

Learn how to delete uploaded static workspace files.

- [Deleting a Static Workspace File](#)
- [Deleting All Static Workspace Files](#)

Deleting a Static Workspace File

To delete an uploaded static workspace file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Workspace Files**.
The Static Workspace Files page appears.
3. If not already selected, click the **View Report** icon.
A report appears. To customize the report, use the Search bar at the top of the page.
4. Select the file to be deleted.
5. On the Edit page, click **Delete**.



See Also:

["About the Search Bar"](#)

Deleting All Static Workspace Files

To delete all uploaded static workspace files:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Files, select **Static Workspace Files**.
The Static Workspace Files page appears.
3. Click **Delete All Files**.

Managing Application-Level Items

Application-level items do not display, but are used as global variables to the application. You can use an application item as a global variable.

- [Creating an Application-Level Item](#)
Create a application-level item on the shared Shared Components page.
- [Viewing Application Item Usage](#)
You can view a list of where application items are used by clicking the **Utilization** tab at the top of the Application Items page.
- [Editing Application-Level Item Attributes](#)
Once you create an application-level item, you can edit it on the Create/Edit Application Item page.
- [About Creating an Application-Level Item that Functions as a Global Variable](#)
You can create a global variable by creating an application-level item with the same name across all applications and setting the scope to Global.



See Also:

- ["About the Differences Between Page Items and Application Items"](#)
- ["Referencing Item Values"](#)

Creating an Application-Level Item

Create a application-level item on the shared Shared Components page.

To create an application-level item:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Items**.
The Application Items page appears.
3. To create an application item, click **Create**.
The Create/Edit page appears.
4. For Name:
 - a. Name - Enter a name for the application-level item.
 - b. Scope - Options include:
 - Global - Select this option if the application shares session state with other applications in the same workspace.

- Application - Select this option if the application does share session state with another application.
5. For Security, Datatype - Select datatype for this item.
 6. For Security, Session State Protection - Select session state protection for this item. Options include:
 - **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.

 **Note:**

If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).

- **Restricted - May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this option when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is only applicable only to items that cannot be used as data entry items and is always observed even if Session State Protection is disabled. This attribute may be used for application items or for page items with any of these Display As types:
 - Display Only (Save State=No)
 - Display Only (Save State=No)
 - Stop and Start Grid Layout (Displays label only)
 - **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. A user-level checksum or a session-level checksum will also suffice (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. A session-level checksum will also suffice (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.
7. For Configuration, Build Option - Select a build option for this component. Build options are predefined settings that determine whether components within an application are enabled.
 8. In Comments, enter any notes regarding this item.

9. Click **Create Application Item**.



See Also:

"About Creating an Application-Level Item that Functions as a Global Variable"

Viewing Application Item Usage

You can view a list of where application items are used by clicking the **Utilization** tab at the top of the Application Items page.

To view the Utilization report:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Items**.
The Application Items page appears.
3. Click the **Utilization** tab at the top of the page.

Editing Application-Level Item Attributes

Once you create an application-level item, you can edit it on the Create/Edit Application Item page.

To edit application-level item attributes:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Items**.
The Application Items page appears.
3. Select an application item.
The Create/Edit page appears.
4. Edit the attributes.
To learn more about an attribute, see field-level Help
5. Click **Apply Changes**.

 **See Also:**

- ["Viewing Field-Level Help"](#)
- ["Understanding Cross-Site Scripting Protection"](#)

About Creating an Application-Level Item that Functions as a Global Variable

You can create a global variable by creating an application-level item with the same name across all applications and setting the scope to Global.

Large applications are often separated into several physical applications that reside in the same workspace. These applications often share session state by configuring the same cookie name in each application.

Another requirement of this type of architecture is the need to share a few global variables. While the number global variables should be kept small, global variables function as an effective means to facilitate communicate between the applications. For example:

- Properties of the current user (such as email address, employee number, and so on).
- Current company in a multitenant application.

To create an application-level item that functions as a global variable:

1. Create an application-level item with the same name in each application to share session state.
2. For Scope, select **Global**.
3. For Security, Session State Protection, select **Restricted**.

 **See Also:**

["Creating an Application-Level Item"](#)

Creating Lists

Add a shared collection of links (or a list) to a database application. To add a list to a page, create a region and specify the region type as List. You control the appearance of a list through list templates.

- [About Static and Dynamic Lists](#)
You can create two types of lists in Oracle Application Express: Static Lists and Dynamic Lists.
- [Creating Static Lists](#)
A Static List is based on predefined display and return values.

- [Creating Dynamic Lists](#)
A Dynamic List is based on a SQL query or a PL/SQL function executed at runtime.
- [Copying a List](#)
You can copy a list from another application or from a list in the existing application.
- [Adding a List to a Page](#)
Once you create a list and list entries, the next step is to add it to a page by creating a list region.
- [Editing List Attributes](#)
Once you create a list, you can edit it on the Lists page.
- [Accessing List Reports](#)
You can view the Unused, Conditional Entries, Utilization, and History reports by clicking the appropriate tab at the top of the Lists page. Note that these reports only display after you create a list.
- [Managing Navigation Menus](#)
Applications using newer themes, such as *Universal Theme - 42*, provide navigation with navigation menus.
- [Managing Navigation Bar Lists](#)
Applications using newer themes, such as *Universal Theme - 42*, include navigation bar lists.

 **See Also:**

- ["Creating a New Template"](#)
- ["List Templates"](#)

About Static and Dynamic Lists

You can create two types of lists in Oracle Application Express: Static Lists and Dynamic Lists.

A **Static List** is based on predefined display and return values. When you create a static list you define a list entry label and a target (either a page or URL). You can add list entries when you create the list (creating from scratch), by copying existing entries, or by adding the list entries. You can control when list entries display by defining display conditions.

A **Dynamic list** is based on a SQL query or a PL/SQL function executed at runtime. A dynamic list enables you to dynamically create styled list items that support mobile frameworks.

The list definition displays a specific type of page item, such as progress bars, sidebar, bullet navigation list, or navigation menu. You can control how a list displays through templates.

**See Also:**

["Understanding Dynamic List Syntax"](#)

Creating Static Lists

A Static List is based on predefined display and return values.

- [About the Process of Creating a Static List](#)
- [Creating a Static List from Scratch](#)
- [About Adding Entries and Sublists to a Static List](#)
- [Adding Entries or Sublists from Scratch](#)
- [Copying Static List Entries Between Lists](#)
- [Reparenting Static List Entries](#)
- [Managing Orphaned Static List Entries](#)
- [Resequencing Static List Entries](#)

About the Process of Creating a Static List

The process of creating a static list involves the following steps:

- **Step 1:** Create the list by running the Create List Wizard.
- **Step 2:** Specify whether to create the list from scratch or by copying an existing list.

You can add list entries by either creating them from scratch or by copying entries from an existing list. If you copy an entire list, you also copy all of its list entries.

- **Step 3:** If creating a list from scratch, you are prompted to select a list type. Select **Static**.

Each list element has a display condition, which enables you to control when it displays. You can define a list element to be either *current* or *non-current* for a specific page.

- **Step 4:** Add the list to a page by creating a List region.

How you perform these steps is a personal preference. You can perform the first three steps when you run the Create List Wizard. Alternatively, you can perform these steps one at a time.

**See Also:**

- ["Adding a List to a Page"](#)
- ["Editing List Attributes"](#)
- ["Accessing List Reports"](#)

Creating a Static List from Scratch

To create a list from scratch:

1. Access the Create/Edit Lists Wizard:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
 - d. Under Navigation, click **Lists**.
 - e. Click **Create**.
2. For Source, select **From Scratch** and click **Next**.
3. For Name and Type:
 - a. Name - Enter a numeric or alphanumeric name for the list.
 - b. Type - Select **Static**.
 - c. Build Option - If applicable, select a build option for this component. Build options are predefined settings that determine whether components within an application are enabled.
 - d. Click **Next**.
4. For Define List Entries, specify the following:
 - a. List Entry Label - Enter a numeric or alphanumeric name for the list.
 - b. Target Page ID or custom URL - Select a target page or enter a custom URL.
 - c. Click **Next**.
5. For Confirm, specify the following:
 - a. Create List Regions - Select whether to create a list region. Options include:
 - **Do not create list region(s)**
 - **Create list region on current page**
 - **Create list region for each target page**
 - b. Region Position - If creating a new region, select the region position.
 - c. Region Template - If creating a new region, select a region template.
 - d. List Template - Select the list template to use to control the appearance of your list.
 - e. Click **Create List**.

About Adding Entries and Sublists to a Static List

Once you create a list, you must add entries to it. You can add list entries when creating a list from scratch, copying a list entry from within a list, or copying existing entries from one list to another.

You can also create hierarchical lists that contain sublists. To create a hierarchical list, you must:

- Select a list template that supports hierarchical lists. To determine which list templates support hierarchical lists, look for templates having the naming convention "with Sublist."
- Select a Parent List Item when you create each list entry.



See Also:

- ["Creating Custom Themes"](#)
- ["List Templates"](#)

Adding Entries or Sublists from Scratch

To add an entry or sublist from scratch:

1. Navigate to the Lists page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
 - d. Under Navigation, select **Lists**.
The Lists page appears.
2. Select a list.
The List Details page appears.
3. Click **Create Entry**.
4. Under Entry:
 - a. Parent List Entry - If applicable, select the parent for this list entry. Use this attribute if you are creating a hierarchical list that contains a sublist.
Note this attribute does not appear if you are creating the parent list entry.
 - b. Sequence - Indicate the order in which list the list entry appears. The sequence determines the order of evaluation.
 - c. Image/Class - Identify the image filename or Font Awesome css class to be used to display this list entry. Control over this attribute is provided by list templates.
 - d. Attributes - Identify the image attributes (such as `width="12" height="12"`) for the list element image.
Use the `#LIST_LABEL#` substitution string to reference the list label text. This substitution string enables the title image attribute to be automatically set based on the value of the list label text. For example:

```
title="#LIST_LABEL#"
```
 - e. Alt Attribute - Identify the image alt attribute for the list element image.
Use the `#LIST_LABEL#` substitution string to reference the list label text. This substitution string enables the Alt attribute to be automatically set based on the value of the list label text. For example:

```
...alt="#LIST_LABEL#"...
```

- f. List Entry Label - Enter the label text for this link.

5. Specify a target type. Options include: **URL** or **Page in this Application**.

If the target location is a **URL**, specify the following:

- a. Target Type - Select **URL**.
- b. URL Target - Enter a URL. For example:

```
http://www.example.com
```

If the target location is a page:

- a. Target Type - Select **Page in this Application**.
- b. Page - Specify the target page number.

Other options include:

- **reset pagination for this page** - Select this option to have the page to return to the first set of data meeting a user's new query.
 - **Printer Friendly** - This option displays the target page using the application's Printer Friendly template. Printer friendly templates optimize a page for printing.
- c. Request - Enter text to set the built-in application item `REQUEST`. This is also the item that is set with the name of a button that was clicked.
 - d. Clear Cache - Enter the page numbers, separated by commas, for which you would like the user's session state to be cleared. To learn more, see field-level Help.
 - e. To set session state (that is, give a listed item a value):
 - Set these items - Enter a comma-delimited list of item names for which you would like to set session state.
 - With these values - Enter a comma-delimited list of values for the items specified in the previous step.

You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, `\1234,56\`).

6. Under Current List Entry:

- a. List Entry Current for Pages Type - Specify when this list entry should be current based on the page type.

List items can be current or non-current. Current list items use the current template; non-current list items use the non-current list item template. The actual condition and templates are defined in subsequent attributes.

- b. List Entry Current for Condition - Displays conditionally. Based on the selection above, define a condition to evaluate. When this condition is true, then the list item becomes current. To learn more, see field-level Help.

7. Under Conditions, specify the appropriate information to make the list entry conditional.

8. Under Authorization, you can specify an authorization scheme.

This authorization scheme must evaluate to TRUE in order for this component to be rendered or otherwise processed.

9. Under Configuration, select a build option for this component.
Build options are predefined settings that determine whether components within an application are enabled.
10. Under Click Counting, specify if you want the list entries to be included in the click count.
If this is a link to an external page, you can count clicks.
11. In User Defined Attributes, specify additional attributes. For example, the following adds a tabindex and accesskey.

```
tabindex="15" accesskey="D"
```
12. When you are finished defining list attributes, click **Create** or **Create and Create Another**.



See Also:

- ["Editing Themes"](#)
- ["Optimizing a Page for Printing"](#)
- "COUNT_CLICK Procedure" in *Oracle Application Express API Reference*

Copying Static List Entries Between Lists

You can copy static list entries from one list to another.

To copy list entries between lists:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Lists**.
The Lists page appears.
2. Select the list to copy.
3. On the Tasks list, click **Copy List Entries from one List to Another**.
4. From Copy List Entries, select the list to copy to and click **Next**.
5. Click **Copy List Entries**.

Reparenting Static List Entries

Use the Reparent List Entries page to manage list entry hierarchy.

To reparent list entries:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Lists**.The Lists page appears.
2. Select the list.
3. From the Tasks list, click **Reparent List Entries within this List**.
4. Use Start With to restrict your view to a subset of hierarchy. Make a selection and click **Go**.
5. From Reparent To, select a new parent.
6. Select the entries to move.
7. Click **Reparent Checked Entries**.

Managing Orphaned Static List Entries

An orphaned list entry is a list entry which has a parent, but the parent is no longer a member of the current list. When a list entry becomes orphaned, remove the parent entry.

To manage orphaned list entries:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Lists**.The Lists page appears.
2. Select the list.
3. From the Tasks list, click **Manage Orphaned List Entries**.
4. Select the entries to remove.
5. Click **Clear Parent Entries**.

Resequencing Static List Entries

You can resequence list entries in increments of 10 on the Clean Up page.

To clean up list entries:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.

- b. Click the **App Builder** icon.
- c. Select an application.
- d. On the Application home page, click **Shared Components**.
- e. Under Navigation, click **Lists**.

The Lists page appears.

2. Select the list.
3. From the Tasks list, click **Clean Up List Entries**.
4. Click **Clean Up**.

Creating Dynamic Lists

A Dynamic List is based on a SQL query or a PL/SQL function executed at runtime.

- [About the Process of Creating a Dynamic List](#)
- [Understanding Dynamic List Syntax](#)
- [Creating a Dynamic List from Scratch](#)

About the Process of Creating a Dynamic List

A dynamic list enables you to create a List component based on items from a SQL query or a PL/SQL function returning a SQL query. These dynamic lists can then be rendered on a page using any List Template from your theme.

The process of creating a dynamic list involves the following steps:

- **Step 1:** Create the list by running the Create List Wizard.
- **Step 2:** Specify whether to create the list from scratch or by copying an existing list.
- **Step 3:** If creating a list from scratch, you are prompted to select a list type. Select **Dynamic**.
- **Step 4:** Enter a SQL query or a PL/SQL function returning a SQL query.
- **Step 5:** Add the list to a page by creating a List region.

See Also:

- ["Adding a List to a Page"](#)
- ["Editing List Attributes"](#)
- ["Accessing List Reports"](#)

Understanding Dynamic List Syntax

You create a dynamic list using a SQL query or a function returning a SQL Query.

Syntax for a SQL query:

```

SELECT level, labelValue label,
       [targetValue]      target,
       [is_current]      is_current_list_entry,
       [imageValue]      image,
       [imageAttributeValue] image_attribute,
       [imageAltValue]   image_alt_attribute,
       [attribute1]      attribute1,
       [attribute2]      attribute2,
       [attribute3]      attribute3,
       [attribute4]      attribute4,
       [attribute5]      attribute5,
       [attribute6]      attribute6,
       [attribute7]      attribute7,
       [attribute8]      attribute8,
       [attribute9]      attribute9,
       [attribute10]     attribute10
FROM ...
WHERE ...
ORDER BY ...

```

Syntax for a function returning a SQL query:

```

RETURN
'SELECT level, labelValue label,'||
'   [targetValue]      target,'||
'   [is_current]      is_current_list_entry,'||
'   [imageValue]      image,'||
'   [imageAttributeValue] image_attribute, '||
'   [imageAltValue]   image_alt_attribute,'||
'   [attribute1]      attribute1,'||
'   [attribute2]      attribute2,'||
'   [attribute3]      attribute3,'||
'   [attribute4]      attribute4,'||
'   [attribute5]      attribute5,'||
'   [attribute6]      attribute6,'||
'   [attribute7]      attribute7,'||
'   [attribute8]      attribute8,'||
'   [attribute9]      attribute9,'||
'   [attribute10]     attribute10' ||
'FROM ...' ||
'WHERE ...' ||
'ORDER BY ...';

```

Where:

- *level* and *labelvalue* are required.
- *level* - For hierarchical lists, the *level* parameter should be supplied. For non-hierarchical lists, this parameter can be set to `NULL`.
- *labelvalue* - Text to appear as list entry.
- *targetvalue* - Target URL to branch to when list entry is selected.
- *is_current* - Controls the behavior of the list entry. Values include:
 - `NULL` - Currency of target is based upon Target URL.
 - `'YES'` - List entry is always current.
 - `'NO'` - List entry is not current.
- *imagevalue* - The name of image to be display on the list entry

- *imageattributevalue* - Attributes of the image, such as the width or height
- *imagealtvalue* - Value for Image ALT tag, required for accessibility purposes in templates where the user must click the image.
- *attribute1* to *10*: These attributes tie in with the existing ten User Attributes exposed on the Static List Entry page.

Creating a Dynamic List from Scratch

To create a dynamic list:

1. Access the Create/Edit Lists Wizard:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
 - d. Under Navigation, click **Lists**.
 - e. Click **Create**.
2. Click **From Scratch** and click **Next**.
3. For Name and Type:
 - a. Name - Enter a numeric or alphanumeric name for the list.
 - b. Type - Select **Dynamic**.
 - c. Build Option - If applicable, select a build option for this component. Build options are predefined settings that determine whether components within an application are enabled.
 - d. Click **Next**.
4. For Query or Static Values:
 - a. Query Source Type - Select **SQL Query** or **Function Returning SQL Query**.
 - b. Query - Enter a SQL query or function returning a SQL query. To view SQL Query examples, expand the Examples region.

Build Query - Click **Build Query** to use a wizard to build a query on the table or view you select. Follow the on-screen instructions.
 - c. Click **Next**.
5. For Create List, specify whether to create a list region:
 - a. If you select Create list region on current page, the following attribute display:
 - Region Position - If creating a new region, select the region position.
 - Region Template - If creating a new region, select a region template.
 - List Template - If creating a new region, select a list template.
 - b. Region Position - If creating a new region, select the region position.
 - c. Region Template - If creating a new region, select a region template.
 - d. Click **Create**.

**See Also:**

["Understanding Dynamic List Syntax"](#)

Copying a List

You can copy a list from another application or from a list in the existing application.

- [Copying a List from the Current Application](#)
- [Copying a List Between Applications](#)

Copying a List from the Current Application

You can copy list entries from one list to another.

To copy list entries between lists:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Lists**.The Lists page appears.
2. Select the list to copy.
3. On the Tasks list, click **Copy List**.
4. On Copy List, select **List in this application** and click **Next**.
5. For Identify List:
 - a. Copy List - Select the list to be copied.
 - b. New List Name - Enter a new name for the copied list.
 - c. Click **Copy**.

Copying a List Between Applications

You can copy a list from the current application or from another application.

To copy a list:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Lists**.

The Lists page appears.

2. Select the list to copy.
3. On the Tasks list, click **Copy List**.
4. On Copy List, select **List in another application** and click **Next**.
5. For Identify List:
 - a. Copy From Application - Select the application from which you want to copy a list.
 - b. Copy List - Select the list you are copying.
 - c. New List Name - Enter a name for the new list.
 - d. Click **Copy**.

Adding a List to a Page

Once you create a list and list entries, the next step is to add it to a page by creating a list region.

To add a list to a page:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. Create a new list region:
 - a. In the Gallery at the bottom of the central pane, click **Regions** and locate **List**.
 - b. Right-click **List** and from the context menu, select **Add To** and then the location.

Tip:

In addition using the context menus, you can also use your mouse to drag and drop components to the appropriate location.

3. In the Property Editor, edit the appropriate attributes:

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Identification:
 - a. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.
 - b. Identification, Type - Select **List**.

5. Source, List - Select the source for this list.
6. Layout:
 - a. Layout, Sequence - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.
 - b. Layout, Parent Region - Select the parent region to which this region belongs. If a parent region is selected then this region is rendered completely inside the parent region..
 - c. Layout, Position - Identify a display point for this region.
7. Appearance, Template - Choose a template to control the look of the region.
8. Server-side Condition, Type - Optionally select a condition type from the list that must be met in order for this component to be rendered or processed. Additional attributes appear based on your selection.
9. Security, Authorization Scheme - Optionally select an authorization scheme to control this component. The authorization must evaluate to TRUE in order for this component to be rendered or executed.
10. Click **Save**.

Repeat these procedures for each page where you would like to add a list.

 **See Also:**

- ["Understanding Conditional Rendering and Processing"](#)
- ["Providing Security Through Authorization"](#)
- ["Creating a New Template"](#)
- ["List Templates"](#) for information about altering list display

Editing List Attributes

Once you create a list, you can edit it on the Lists page.

To edit a list:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Lists**.

The Lists page appears.

2. Select a list.

The List Details page appears.

3. To change the list name or build options, click the **Edit List** button.
4. To change the list name or build options:
 - a. Click the **Edit List** button.
 - b. Edit the appropriate attributes.
 - c. Click **Apply Changes**.
5. To edit the list sequence, link text, target, or current status for all list entries, click **Grid Edit**.
 - a. To edit the list sequence, link text, target, or current status for all list entries, click **Grid Edit**.
 - b. Edit the appropriate attributes.
 - c. Click **Apply Changes**.

Accessing List Reports

You can view the Unused, Conditional Entries, Utilization, and History reports by clicking the appropriate tab at the top of the Lists page. Note that these reports only display after you create a list.

To view list reports:

1. Navigate to the Lists page.
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. On the Application home page, click **Shared Components**.
 - e. Under Navigation, click **Lists**.The Lists page appears.
2. Click the appropriate tab:
 - **Unused** - Click the **Unused** tab to identify lists that are not used in the current application.
 - **Conditional Entries** - Click the **Conditional Entries** tab to view conditional lists.
 - **Utilization** - Click the **List Utilization** tab to access the Utilization report. This report displays all lists included in the current application. From the report, you can:
 - Edit list entries by selecting the list name.
 - View the pages on which the list appears by clicking the number in the Pages column.
 - View the template used with the list by expanding **List Template Utilization** and then clicking the name to view or edit the list template.
 - **History** - Click the **History** tab to view changes to list definitions and list entries by developer and date.

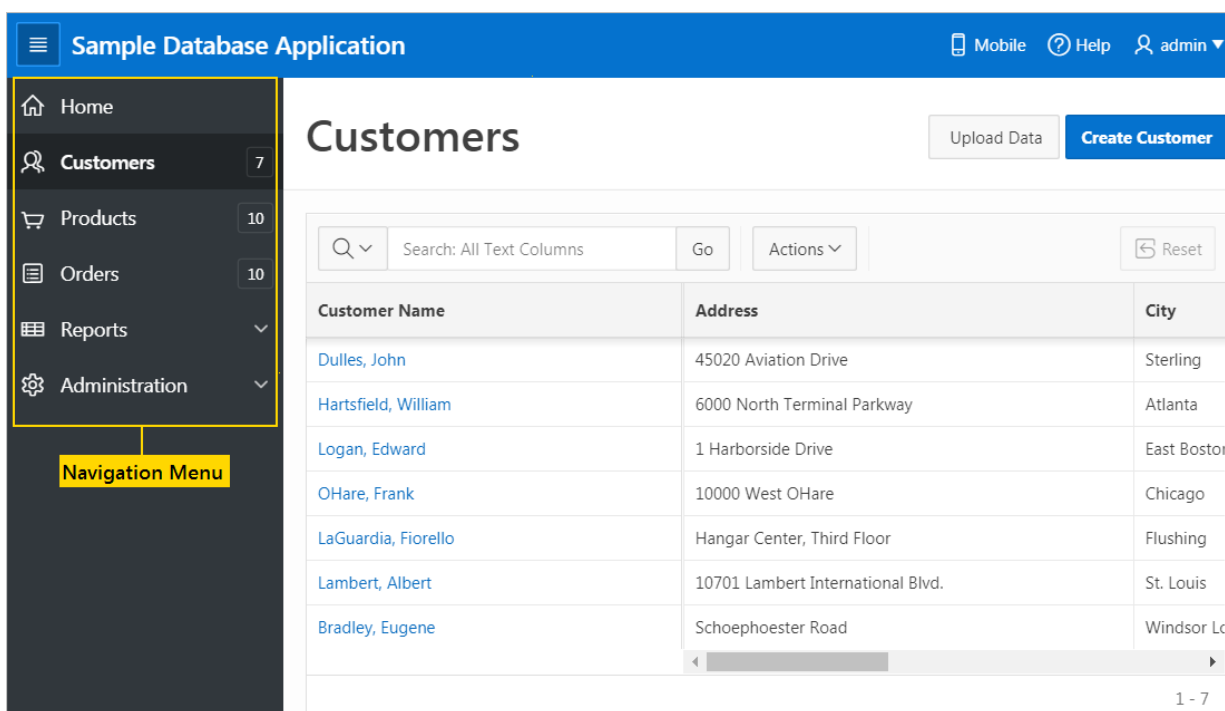
Managing Navigation Menus

Applications using newer themes, such as *Universal Theme - 42*, provide navigation with navigation menus.

- [About Navigation Menus](#)
- [Editing Navigation Menu Lists](#)
- [Changing Where and How Navigation Menus Display](#)
- [About Switching from Tabs to Navigation Menu](#)

About Navigation Menus

A **navigation menu** is list with hierarchical list entries. When you create an application, the Create Application Wizard automatically creates a navigation menu for you and populates it with list entries linking to the application pages you have created. The example below shows the *Sample Database Application* with a navigation menu displaying as a sidebar. In this example, the navigation menu includes list entries for the application pages: Home ; Customers; Orders; Reports; and Administration.



The screenshot displays the 'Sample Database Application' interface. On the left, a dark sidebar navigation menu is highlighted with a yellow box and labeled 'Navigation Menu'. The menu items are: Home, Customers (7), Products (10), Orders (10), Reports, and Administration. The main content area shows the 'Customers' page with a search bar, a table of customer records, and a 'Create Customer' button.

| Customer Name | Address | City |
|---------------------|-----------------------------------|-------------|
| Dulles, John | 45020 Aviation Drive | Sterling |
| Hartsfield, William | 6000 North Terminal Parkway | Atlanta |
| Logan, Edward | 1 Harborside Drive | East Boston |
| OHare, Frank | 10000 West OHare | Chicago |
| LaGuardia, Fiorello | Hangar Center, Third Floor | Flushing |
| Lambert, Albert | 10701 Lambert International Blvd. | St. Louis |
| Bradley, Eugene | Schoephoester Road | Windsor Lc |

Editing Navigation Menu Lists

Since a Navigation menu is a list, you create and edit in the same way as any other list.

You can access Navigation menus in Shared Components in the Navigation region in two ways:

- Select the application and then click **Shared Components**. Under the Navigation region, select **Navigation Menu**.
- Select the application and then click **Shared Components**. Under the Navigation region, select **Lists**.

Once you select the Navigation Menu, you edit it as you would any other list.



See Also:

["Creating Lists"](#)

Changing Where and How Navigation Menus Display

Navigation menus can display either as a side bar or at the top of the window. Navigation menus displaying as a side bar are responsive. In other words, based on the available space, the navigation menu displays a full menu or collapses to a narrow icon bar. Navigation menus are controlled at the theme and user interface level. You can change how and where a navigation menu displays by editing the application User Interface Details

To edit the Navigation menu attributes in User Interface Details:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.

The Application home page appears.

3. Access the User Interfaces page:
 - a. Click **Shared Components**.
 - b. Under User Interface, click **User Interface Attributes**.

The User Interfaces page appears. Defined User Interfaces display in the **User Interfaces** region at the top of the page.

4. To edit attributes for a specific user interface, click the **Edit** icon adjacent to the User Interface.

The User Interface Details page appears.

5. Under Navigation Menu, edit the following attributes.
 - Display Navigation - Turn navigation on or off by selecting **Yes** or **No**.
 - Navigation Menu List - Select the list utilized for the navigation menu for the application.
 - Position - Select the position where you would like your navigation menu to be placed in this application.
 - List Template - Select the List Template used to render the navigation menu for the application.
 - Template Options - Set Template Options for the List Template used for the navigation menu list for the application.

To learn more about an attribute, see field-level Help.

6. Click **Apply Changes** to save your changes.

7. Run the application to view your changes.

 **Note:**

["Managing the Application User Interface"](#)

About Switching from Tabs to Navigation Menus

You can switch from a theme using tabs to a theme using navigation menus if the existing application used one level of tabs was used. If an existing applications uses two levels of tabs, you must convert it to one level before changing the theme.

 **See Also:**

["Editing Themes"](#)

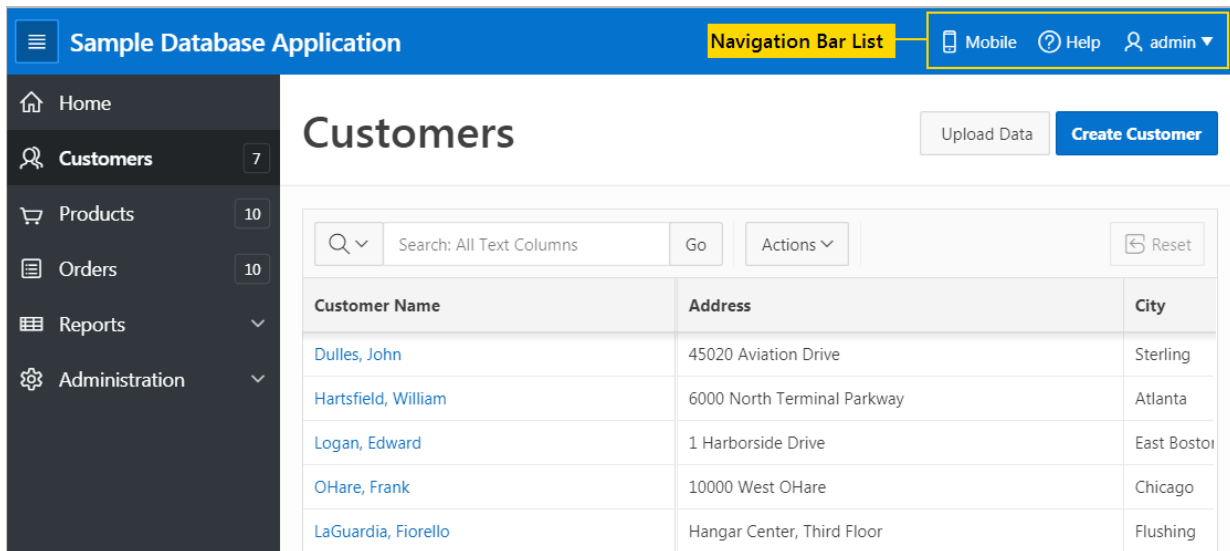
Managing Navigation Bar Lists

Applications using newer themes, such as *Universal Theme - 42*, include navigation bar lists.

- [About Navigation Bars](#)
- [Editing Navigation Bar Lists](#)
- [Editing Navigation Bar User Interface Details](#)

About Navigation Bars

A **navigation bar list** displays with a list template in the `#NAVIGATION_BAR#` position on your page template. The example below shows the *Sample Database Application* with a navigation bar displaying in the upper right corner.



The screenshot shows a web application interface for a 'Sample Database Application'. The top navigation bar is blue and contains a hamburger menu icon, the application name, a yellow box labeled 'Navigation Bar List', and user options for Mobile, Help, and admin. A left sidebar lists navigation items: Home, Customers (7), Products (10), Orders (10), Reports, and Administration. The main content area is titled 'Customers' and includes 'Upload Data' and 'Create Customer' buttons. Below is a search bar and a table of customer records.

| Customer Name | Address | City |
|---------------------|-----------------------------|-------------|
| Dulles, John | 45020 Aviation Drive | Sterling |
| Hartsfield, William | 6000 North Terminal Parkway | Atlanta |
| Logan, Edward | 1 Harborside Drive | East Boston |
| O'Hare, Frank | 10000 West O'Hare | Chicago |
| LaGuardia, Fiorello | Hangar Center, Third Floor | Flushing |

Editing Navigation Bar Lists

Since a navigation bar is a list, you create and edit in the same way as any other list.

You can access navigation bars in Shared Components in the Navigation region in two ways:

- Select the application and then click **Shared Components**. Under the Navigation region, select **Navigation Bar List**.
- Select the application and then click **Shared Components**. Under the Navigation region, select **Lists**.

Once you select the navigation bar list, you edit it as you would any other list.



See Also:

"Creating Lists"

Editing Navigation Bar User Interface Details

Navigation Bar User Interface Details settings enable you to select list and list templates. Selecting classic implementation uses tabs instead of a list.

To edit the Navigation Bar User Interface Details:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
The Application home page appears.
3. Access the User Interfaces page:
 - a. Click **Shared Components**.
 - b. Under User Interface, click **User Interface Attributes**.

The User Interfaces page appears. Defined User Interfaces display in the **User Interfaces** region at the top of the page.

4. To edit attributes for a specific user interface, click the **Edit** icon adjacent to the User Interface.

The User Interface Details page appears.

5. Under Navigation Bar, edit the following attributes:
 - Implementation - Select how to you wish to implement the navigation bar in this application. Options include:
 - **Classic** renders the navigation bar as a classic navigation bar in the #NAVIGATION_BAR# position on your page template.
 - **List** renders the navigation bar as a list, using the selected list and list template in the #NAVIGATION_BAR# position on your page template.
 - Navigation Bar List - Select the list utilized for the navigation bar on this application.
 - List Template - Select the List Template used to render the navigation menu for the application.
 - Template Options - Set Template Options for the List Template used for the navigation menu list for the application.
6. Click **Apply Changes** to save your changes.
7. Run the application again to view your changes.

 **Note:**

["Managing the Application User Interface"](#)

Managing Report Output

Learn how to print a report region by defining a report query and how to use a report layout to format a report region.

- [Printing a Report Region by Defining a Report Query](#)
- [Formatting a Report Region or Report Query Using Report Layouts](#)

Printing a Report Region by Defining a Report Query

Print a report region by defining a report query.

- [About Report Queries](#)
- [Creating a Report Query](#)
- [Editing a Report Query](#)
- [Copying a Report Query](#)

About Report Queries

You can print a report region by defining a report query as a Shared Component. A report query identifies the data to be extracted. Unlike SQL statements contained in regions, report queries contain SQL statements that are validated when you save the query. Note that report queries must be SQL statements, not functions returning SQL statements.

You can associate a report query with a report layout and download it as a formatted document. If no report layout is selected, a generic layout is used. The generic layout is intended to be used to test and verify a report query. When using the generic layout option and multiple source queries are defined, only the first result set is included in the print document. The reports can include session state of the current application.

To make these reports available to end users, you then integrate them with an application. For example, you can associate a report query with a button, list item, branch, or other navigational component that enables you to use URLs as targets. Selecting that item then initiates the printing process.

Creating a Report Query

Note that the availability of the report query options depends on how your service administrator configures report printing for your instance. All options described in these steps may not be available to you.

To create a report query:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Reports, click **Report Queries**.
3. Click **Create**.
4. For Query, specify the following information:
 - a. Report Query Name - Enter a name to identify the report query. When referencing a report query as a link target, the report query name is part of the request string.
 - b. Output Format - Select the report output format. Valid options include:
 - PDF - Adobe Portable Document Format.
 - Word - Microsoft Word Rich Text Format.
 - Excel - Microsoft Excel format. Note that this is not a true .xls file because the content is HTML-based.
 - HTML
 - XML - Extensible Markup Language.
 - c. Item - Select the item to hold the format value. You can use this item to determine the output format at runtime.

- d. View File As - Select how the web browser opens the report query document. Options include:
 - **Attachment** - Displays a File Open/Save dialog box when the user clicks the report query download button.
 - **Inline** - Displays the report query document inside the Browser window.
 - e. Session State - Select this check box to reference session state values, such as the values of page items in the XML structure of your report.
 - f. Click **Next**.
5. For SQL Query, enter a SQL statement directly or click **Query Builder** to build a SQL statement by clicking and pointing.

To reference applications and page items in the SQL statement, reference them as bind variables. Click **Set Bind Variables** to be able to enter in values for each bind variable and test the query to ensure the expected output is returned.

6. For Download Definition, specify the following:
- a. Data Source for Report Layout - Use this option to download a report definition export file.

 **Tip:**

The report query definition is what a developer would use as the source for tools like BI Publisher Desktop or Altova Stylevision to create the report layout. Some tools accept both an XML export of the data as well as an XML schema definitions as source, other tools may only accept one or the other. If the source query does not return data at design time, then XML schema provides a way to export the definition without data.

Select **XML Data** or **XML Schema** and click **Download**.

XML Schema creates XSD file (that is, a XML schema definition of your XML structure.)

- b. Add Query - Click this button to add another query to the report query. Adding another query enables the production of complex reports with more than one output component (for example, a chart and a report based on two different queries).
 - c. Create Report Query - Click this button to exit the wizard early. The Report Layout Source defaults to use the generic report layout.
 - d. Click **Next**.
7. For Upload Report Layout, specify the following:
- a. Report Layout Source - Specify how the report layout is derived. If you choose the report based file layout then the layout name and report layout file must be provided.
 - b. Layout Name - Enter a layout name (optional). The default is the report query name.
 - c. Report Layout File - Click **Browse** and select a Rich Text Format (RTF).

- d. Click **Next**.
8. For Confirm:
 - a. Query Name - Identifies the query.
 - b. Report Layout - Identifies the report layout you selected.
 - c. Output Format - Identifies the format for this report query.
 - d. Derive from Item - Identifies the item to hold the format information.
 - e. URL - To integrate this report with your application, use the displayed URL as the target for a button, list item, link, or other navigational component. End users can click a button, for example, to start the printing process.
 - f. Test Report - Click this to preview your report. If you have chosen to use a generic report layout then the Test Report does not produce an output.
 - g. Click **Finish**.

The Report Query is created and saved to Shared Components.



See Also:

["About Report Printing Configuration Options"](#)

Editing a Report Query

To edit a report query:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.
2. Under Reports, click **Report Queries**.
3. Select the appropriate report query.
4. On the Edit page, edit the appropriate attributes.
5. Click **Apply Changes**.

Copying a Report Query

To copy a report query:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Reports, click **Report Queries**.
3. On the Report Queries page, click **Copy**.
4. On the Copy Report Query, select the query you want to copy, enter a name for the report query, and click **Copy**.

The copy appears in the query list.

Formatting a Report Region or Report Query Using Report Layouts

Create a report layout to format a report region or report layout.

- [About Report Layouts](#)
- [About Report Layout Options](#)
- [Creating a Report Layout](#)
- [Editing a Report Layout](#)
- [Copying a Report Layout](#)

About Report Layouts

To format either a classic report region or report query, you associate it with a report layout. Using report layouts renders the data in a printer-friendly format. If you do not select a report layout, a default XSL-FO layout is used. The default XSL-FO layout is always used for rendering Interactive Report regions.

When creating and using report layouts, you can:

- Take advantage of the default layouts for report regions and generic layouts for report queries provided with Oracle Application Express.
- Utilize the built-in XSL-FO-based layouts for report regions by copying and customizing the code. You can edit several attributes for report regions that control page size, fonts, colors, and so on.
- Create RTF or XSL-FO report layouts to customize the report look and feel. To use RTF report layouts, your Oracle Application Express service administrator must select the Advanced setting for your site.

Note:

Interactive reports do not support a custom report layout.

See Also:

"Configuring Report Printing" in *Oracle Application Express Administration Guide*

About Report Layout Options

You can create a report layout based on one of these options:

- **Generic Columns** - A generic report layout works with most query result sets. With this layout, the number of columns is automatically adjusted when generating the printable document.

Many report layout attributes can be defined declaratively for report regions using the built-in XSL-FO default layout. This step allows for creating customizable copies of the built-in default XSL-FO layout, if additional control over the report layout is needed.
- **Named Columns** - A named column report layout is a query-specific report layout designed to work with a defined list of columns in the query result set. This type of layout is used for custom-designed layouts when precise control of the positioning of page items and query columns is required.

Note that the availability of the Report Layout options depends on how your site administrator configured the report printing settings at your site. All options described in these steps may not be available to you.



See Also:

"Configuring Report Printing" in *Oracle Application Express Administration Guide*.

Creating a Report Layout

To create a report layout:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Reports, click **Report Layouts**.
3. Click **Create**.
The Create Report Layout wizard appears.
4. For Report Layout Type, select an option and click **Next**:
 - Generic Columns (XSL-FO) - Uses the default template to populate the report. In the next step, you can customize the default code.
 - Named Columns (XSL-FO) - Requires that you upload an XSL-FO or RTF file in the next step.
5. For Layout Source, review and edit the appropriate information. The options that appear on this page depend on the layout type you select:

- If you selected **Generic Columns**:
 - a. Report Layout Name - Enter a name to identify the report layout when associating it with a report query or report region.
 - b. Report Layout - The report layout is the XSL-FO based definition of the page formatting. All attributes defining page size, orientation, fonts, styles, and so on, are defined in this section.
 - c. Report Column Heading - Defines the look of each cell in the report heading row.
 - d. Report Column - Defines the look of each cell for all report rows.
 - e. Report Column Width - This width is computed at runtime or can be derived from the report column definition of a report region.
 - If you selected Named Columns:
 - a. Layout Name - Enter a name to identify the report layout when associating it with a report query or report region.
 - b. Report Layout File - Upload the file containing the report layout.
- 6. Click **Create Layout**.

Editing a Report Layout

You can edit a generic column report layout directly in Application Express. However, to edit a named column report layout, you must download the current file, edit it, and then upload it again.

To edit a report layout:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.The Shared Components page appears.
2. Under Reports, click **Report Layouts**.
3. On the Report Layouts page, select the layout you want to edit.
4. For generic column layouts, edit the layout directly on the Edit Report Layout page and click **Apply Changes**.
5. For named column layouts, click **Download** and save the file to your computer. Edit the file and then upload the updated version as a new report layout.

Copying a Report Layout

You can copy a report layout to edit and save.

To copy a report layout:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.

- c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

- 2. Under Reports, click **Report Layouts**.
- 3. On the Report Layouts page, click **Copy**.
- 4. On the Copy Report Layout page, select the layout you want to copy, enter a name for the copy, and click **Copy**.

The copy appears in the layout list.

- 5. On Copy Report Layout:
 - a. Copy Report Layout - Select the layout you want to copy.
 - b. Report Layout Name - Enter a name for the copy.
 - c. Click **Copy**.

Managing Application Data

Learn how to manage application data: add data loading capability, use collections to temporarily store information, execute SQL or PL/SQL defined at the component-level on a remote database, manage external web services through Web Source Modules, Remote Servers, and Web Credentials, manage legacy Web services, accessing data with database links, and utilizing Data Manipulation Language (DML) to update or delete rows of a table.

- [Creating Applications with Data Loading Capability](#)
Create applications with data loading capability that enable end users to dynamically import data into a table within any schema to which they have access.
- [About Using Collections](#)
Collections enable you to temporarily capture one or more nonscalar values. You can use collections to store rows and columns currently in session state so they can be accessed, manipulated, or processed during a user's specific session.
- [Managing REST Enabled SQL References](#)
Execute SQL queries or PL/SQL defined at the component-level on a remote database using REST Enabled SQL References.
- [Managing Web Source Modules](#)
Web Source Modules act as a reference to one or multiple external web services. A module can contain one or many Web Source Operations which are the references to a concrete external web service.
- [Managing Remote Servers](#)
A Remote Server is separate entity that stores Web Source server information. Remote Servers can be shared among multiple Web Sources, thus enabling you to information such as the Base URL or Authentication.
- [Managing Web Credentials](#)
Manage secure credentials to connect to REST Enabled SQL or other REST services.
- [Managing Legacy Web Services](#)
Legacy Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment.
- [Accessing Data with Database Links](#)
Access data remotely by creating a database link.
- [About DML Locking](#)
Use automatic Data Manipulation Language (DML) in Oracle Application Express to update or delete rows of a table.

Creating Applications with Data Loading Capability

Create applications with data loading capability that enable end users to dynamically import data into a table within any schema to which they have access.

To import data, end users run a Data Load Wizard that uploads data from a file or copies and pastes data entered by the end user directly into the wizard.

- [About Creating a Data Load Wizard](#)
- [Creating a Data Load Wizard](#)
- [Re-creating Data Load Pages](#)
- [Editing the Data Load Definition](#)

 **See Also:**

"Data Loading Wizard Examples" in *Oracle Application Express End User Guide* to learn about using the Data Load Wizard.

About Creating a Data Load Wizard

To create a Data Load Wizard, an application developer creates a Data Load page with the Create Page Wizard. During the process of creating the Data Load Wizard, developers can specify the upload table and its unique columns, table lookups, and data transformation rules.

 **Note:**

A Data Load Wizard is not designed or intended to load hundreds of thousands of rows of data. While it is possible to use a Data Load Wizard to load this high volume of data, you may encounter performance issues with both transmitting and loading large data files. Tools like Oracle SQL Developer and Oracle SQL*Loader are better suited to loading large volumes of data.

The Data Load Wizard includes support for the following:

- **Table Definitions** - This definition specifies the data upload table name with its unique key columns.
- **Data Transformation Rules** - For formatting transformations such as changing import data to uppercase, lowercase, and so on, you must define data transformation rules. For example, if the import file includes column data with both upper and lowercase and the upload table requires all uppercase, you can define a data transformation rule to insert only uppercase into the target column.
- **Table Lookups** - If data existing in the import file must be mapped to data in another table, specify a table lookup to perform the mapping. For example, if the import file contains a department name for the DEPTNO column but the upload table requires a number for that column, use a table lookup rule to find the corresponding department number for that department name in another table.
- **Column Name Aliases** - There are many situations when a developer does not wish to expose the table column names to the end user, or to expose all columns

to the end user. In those situations, you can create a column aliases for the columns that need to be exposed.

- **Manage Concurrency** - If multiple users are uploading data at the same time, developers can use extra column to track the version of data in the underlying table. The Data Load Wizard can use this column to check and signal the end user if anyone else is working with the same data at the same time. This is particularly important if uploading into a table that is regularly updated.
- **Multiple Spreadsheet Columns** - There are many situations when a spreadsheet to be uploaded has multiple columns that the developer wants to concatenate and upload in to one table column (for example `FirstName` and `LastName` on spreadsheet can be uploaded into `ENAME` of the `EMP` table).
- **Skip Validation** - You can improve data loading performance when uploading a large number of records by skipping the validation step. If uploading thousands of records, the end user might not be interested in validating each record. If you are certain that each record will be inserted as new record, the data loading process does not need to check for duplicates.

The newly generated Data Load Wizard consists of four pages that provide users with the ability to upload data from a file or by copy and paste, define data and table mappings, validate the data, and finally to upload the data to the table. The developer can later edit the Data Load Wizard's definitions such as table lookups and data transformation rules, by accessing Shared Components, Data References, Data Load Definitions.

Supported Data Types

Data Load Wizards support the following data types:

- `VARCHAR2`
- `DATE`
- `TIMESTAMP`
- `NUMBER`

Unsupported Data Types

Data Load Wizards do not support the following data types:


- Large objects (`BLOB` and `CLOB`)
- Complex types (`XMLTYPE` and `SDO_GEOMETRY`)
- `CHAR`

See Also:

- ["Creating a Data Load Wizard"](#)
- ["Editing the Data Load Definition"](#)

Creating a Data Load Wizard

To create a Data Load Wizard:

1. On the Workspace home page, click the **App Builder** icon.
 2. Select the application.
 3. Click **Create Page**.
 4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Data Loading**.
-  **Tip:**

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.
5. For Data Load Table:
 - a. Data Load Definition - Choose whether to create a new or to re-use an existing Data Load definition.
 - b. Definition Name - Enter the name of this data load definition.
 - c. Owner - Select the owner of the table on which the form will be based..
 - d. Table Name - Select the table to use for data loading (also known as the upload table).
 - e. Unique Column 1 - Identify the column name(s) to be used as the primary unique key column during the data load process. You can define up to 3 unique key columns.
 - f. Case Sensitive - Identify whether the selected unique key column is case sensitive. By default, this is set to **No**.
 - g. Define additional Unique Columns. You can define up to 3 unique key columns.
 - h. Click **Next**.
 6. For Add Transformation Rules (optional), specify the following then click **Add**.
 - a. Select Column(s) to create a transformation rule - Select the column on which the transformation rule definition is to be based and move them to the right.
 - b. Rule Name - Enter a name for this transformation rule.
 - c. Sequence - Specify the sequence for the transformation rule. The sequence determines the order of execution.
 - d. Type - Select the type of transformation rule you want to perform.

- e. Provide additional details based on the transformation type you have chosen.
 - f. Click **Add Transformation**.
 - g. Click **Next**.
7. For Table Lookups:
 - a. Add new table lookup for Column (optional) - Identify the column on which the table lookup definition is to be based.
 - b. Lookup Table Owner - Select the owner of the lookup table.
 - c. Lookup Table Name - Identify the table to be used for this table lookup definition.
 - d. Return Column - Select the name of the column returned by the table lookup. This value will be inserted into the load column specified and is generally the key value of the parent in a foreign key relationship (for example: DEPTNO).
 - e. Upload Column - Select the name of the column end users will upload instead of the return column. This is the column that contains the display value from the lookup table (for example: DNAME).
 - f. Upload Column 2 - Select the name of the second column to be uploaded to uniquely identify the return column if necessary. For example, to uniquely identify a State Code it may be necessary to upload the State Name and Country.
 - g. Upload Column 3 - Select the name of the third column to be uploaded to uniquely identify the return column.
 - h. Click **Add lookup** to add the lookup definition. Repeat the previous steps to add additional table lookups.
 - i. Click **Next** finish creating lookups.
8. For Page Attributes:
 - a. Page Name - Enter a page name for each step.
 - b. Page Number - Enter a page number for each step.
 - c. Page Mode - Identify the page mode. To learn more, see field-level Help.
 - d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page and which breadcrumb navigation control you want to use.
 - e. Click **Next**.
9. For Navigation Menu:
 - a. Navigation Preference - Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
10. For Buttons and Branching, specify the branching for the buttons on the data load wizard pages:
 - a. New Button Label - Enter text to display on the Next button.
 - b. Previous Button Label - Enter text to display on the Previous button.
 - c. Cancel Button Label - Enter text to display on the Cancel button.
 - d. Cancel Button Branch to Page - Specify the page number to branch to when the user clicks Cancel.

- e. Finish Button Label - Enter text to display on the Submit button.
 - f. Finish Button Branch to Page - Specify the number of the page to branch to. You can choose to branch back to the same page or any other page in your application.
 - g. Click **Create**.
11. Click **Save and Run Page** to test the Data Load Wizard.

 **Tip:**

After creating Data Load Wizard pages, if you wish to make changes, Oracle recommends re-creating new pages without deleting the data loading definitions as described in the next section.

 **See Also:**

- ["Re-creating Data Load Pages"](#)
- ["Editing Pages in Page Designer"](#)

Re-creating Data Load Pages

If you wish to make changes to your Data Load Wizard pages, Oracle recommends re-creating new pages without deleting the data loading definitions.

To re-create Data Load Wizard pages without deleting the data loading definitions:

1. Navigate to the Data Load Definitions page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
 - d. Under Data References, click **Data Load Definitions**.
2. Click the Data Load Definition you want to re-create.
The Data Load Table Details page appears.
3. From Tasks, click Re-create Data Load Pages.
The Page Attributes page of the Create Data Load Wizard appears.
4. For each page, edit the appropriate attributes and click **Next**.
To learn more about any attributes, see field-level Help.
5. Click **Confirm**.

Editing the Data Load Definition

A Data Load Definition is comprised of a data load table, table rules, and lookup tables used by the Data Load Wizard in your application. A data load table is an existing table in your schema that has been selected for use in the data loading process, to upload data.

To edit a Data Load Definition:

1. Navigate to the Data Load Definitions page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
 - d. Under Data References, click **Data Load Definitions**.
2. Click the Data Load Definition you want to edit.
The Data Load Table Details page appears. To learn more about any attributes, see field-level Help.
3. For Data Load Definition:
 - Name - Name for the Data Load Definition.
 - Data Load Table - Edit the data load table name and add, remove or edit unique columns.
 - Unique column 1 - The column name used as the primary unique key column during the data load process.
 - Case Sensitive - Identify whether the selected unique key column 1 is case sensitive. By default, this is set to No.
 - Unique column 2 - If the unique key definition of the selected table is a compound key, consisting of 2 or more columns, this column name is used as the second unique key column during the data load process.
 - Case Sensitive - Identify whether the selected unique key column 2 is case sensitive. By default, this is set to No.
 - Unique column 3 - If the unique key definition of the selected table is a compound key, consisting of 2 or more columns, this column name is used as the third unique key column during the data load process.
 - Case Sensitive - Identify whether the selected unique key column 3 is case sensitive. By default, this is set to No.
 - Skip Validation - One step in data loading is to validate actions to be taken on records to be uploaded. Select Yes to skip validation.
4. Transformation Rules lists previously defined transformation rules.
 - To create a new rule:
 - a. Click **Create Transformation Rule**.
 - b. Edit the attributes.
To learn more about an attribute, see field-level Help.

- c. Click **Create**.
 - To edit an existing rule:
 - a. Click the rule name.
 - b. Edit the attributes.
 - c. Click **Apply Changes**.
5. Table Lookups map data in the import file to data found in another table.
 - To create a new Table Lookups:
 - a. Click **Create Table Lookup**.
 - b. Edit the attributes.
 - c. Click **Create**.
 - To edit an existing Table Lookup:
 - a. Click column name.
 - b. Edit the attributes.
 - c. Click **Apply Changes**.
6. Column Name Aliases define aliases to help users correctly identify the columns to upload.

To add Column Name Aliases:

 - a. Click **Edit List of Values**.
 - b. Edit the attributes.
 - c. Click **Create**.
7. From Concurrency Column Name, select a column to be used for concurrency management.

Concurrency gives the developer the option to select a column to check the version of the data in the underlying table. This is particularly important if uploading into a table that is regularly updated.
8. Click **Apply Changes**.

About Using Collections

Collections enable you to temporarily capture one or more nonscalar values. You can use collections to store rows and columns currently in session state so they can be accessed, manipulated, or processed during a user's specific session.

You can think of a collection as a bucket in which you temporarily store and name rows of information.

The following are examples of when you might use collections:

- When you are creating a data-entry wizard in which multiple rows of information first need to be collected within a logical transaction. You can use collections to temporarily store the contents of the multiple rows of information, before performing the final step in the wizard when both the physical and logical transactions are completed.

- When your application includes an update page on which a user updates multiple detail rows on one page. The user can make many updates, apply these updates to a collection and then call a final process to apply the changes to the database.
- When you are building a wizard where you are collecting an arbitrary number of attributes. At the end of the wizard, the user then performs a task that takes the information temporarily stored in the collection and applies it to the database.

You insert, update, and delete collection information using the PL/SQL API `APEX_COLLECTION`.



See Also:

"APEX_COLLECTION" in *Oracle Application Express API Reference*

Managing REST Enabled SQL References

Execute SQL queries or PL/SQL defined at the component-level on a remote database using REST Enabled SQL References.

- [About REST Enabled SQL Service](#)
Create REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote Oracle database.
- [Before You Begin: REST Enabled SQL Service Requirements](#)
Prior to creating a REST enabled SQL reference, developers must complete these requirements.
- [Creating a REST Enabled SQL Service Reference](#)
Create REST Enabled SQL service references on the REST Enabled SQL page.
- [Example: Creating a REST Enabled SQL Service Reference](#)
Review an example of creating REST Enabled SQL service reference on a remote database.
- [Editing a REST Enabled SQL Service Reference](#)
Edit existing REST Enabled SQL references on the Edit page.
- [Deleting a REST Enabled SQL Service Reference](#)
Delete REST Enabled SQL references on the Edit page by clicking **Delete**.
- [Creating or Updating Components to Use a REST Enabled SQL](#)
Create or update classic reports, interactive reports, CSS calendars, JET charts, Tree regions, Toggle Column reports, and Reflow Table reports to point to data on a remote database using a REST Enabled SQL reference.

About REST Enabled SQL Service

Create REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote Oracle database.

Because REST Enabled SQL services are stored at the workspace-level within Application Express components, they are available to all applications within a workspace. To use REST Enabled SQL services, the remote database must include

Oracle REST Data Services (ORDS) release 17.3 or later on the front-end and have the REST Enabled SQL feature enabled. All SQL and PL/SQL is sent to the Oracle REST Data Services (ORDS) instance which executes it on the remote database.

How Do REST Enabled SQL Service References Differ from Database Links?

Both REST Enabled SQL services and database links enable developers to access data remotely. However, these features access remote data differently. Key differences between database links and REST Enabled SQL Services include:

- Database Link:
 - Functions at the SQL-level which enables developers to use remote tables and local tables in the same SQL query.
 - Works over SQL*Net, which can be problematic when connecting to a Cloud instance over the internet.
- REST Enabled SQL Service:
 - Functions at the workspace-level. Developers can create an Application Express component with a query on a REST enabled SQL service but cannot join it to a local table.
 - Works with JSON over HTTP(s) which makes it easy-to-use it in Cloud environments or over the internet.

Both Database Links and REST Enabled SQL fetch data over the network which is significantly slower than fetching data from a table in the local database. When evaluating the best approach for your environment, be sure to evaluate the impact on page view performance and always consider replicating remote data in local tables, with an appropriate refresh algorithm.

Exporting and Importing REST Enabled SQL Services

When you export an application, used REST Enabled SQL references are added to the export file. If you export an application and import it into another workspace, Application Express checks whether the target workspace already contains REST Enabled SQL References with the same static ID. If a REST Enabled SQL reference already exists, the application uses the existing reference. If the reference does not exist, it is created in the target workspace.

Before You Begin: REST Enabled SQL Service Requirements

Prior to creating a REST enabled SQL reference, developers must complete these requirements.

Complete the following requirements before creating REST enabled SQL reference:

1. Set up a remote Oracle database.
2. On the remote Oracle database, install Oracle REST Data Services (ORDS) release 17.3 or later.

Tip:

This installation is completely independent from the Oracle REST Data Services used as the Oracle Application Express web server

3. Configure and enable the REST Enabled SQL service feature. See “Configuring REST-Enabled SQL Service Settings” in *Oracle REST Data Services Installation, Configuration, and Development Guide*.
4. Activate REST Enabled SQL for the target schema on the remote database to be accessed by running `ORDS.ENABLE_SCHEMA`.

Log in to the database schema to be enabled for REST Enabled SQL using a "classic" client and execute the following:

```
begin
    ords.enable_schema;
end;
/

commit
/
```

The REST Enabled SQL service is then available with a URL in the following format:

```
http://host:port/ords/schema
```

Where:

- *host* is the name of the system where Oracle REST Data Services is installed.
- *port* is the port number assigned when configuring Oracle REST Data Services. In a default installation, this number is 8080.
- *ords* is the service name defined when configuring Oracle REST Data Services.
- *schema* is the target schema.

 **Tip:**

When providing the URL for the target schema in Oracle Application Express, do not append `/_/sql`.

You will use a URL using the above format when creating the REST Enabled SQL service reference in Oracle Application Express.

For example, the following is a REST Enabled SQL service for the `scott_obe` schema with ORDS running in standalone mode:

```
http://server.example.com:8080/ords/scott_obe
```

Creating a REST Enabled SQL Service Reference

Create REST Enabled SQL service references on the REST Enabled SQL page.

 **Tip:**

Before creating a REST enabled SQL reference, developers must complete the requirements described in "[Before You Begin: REST Enabled SQL Service Requirements](#)".

To create a REST Enabled SQL service reference:

1. Navigate to the REST Enabled SQL page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **REST Enabled SQL**.

 **Tip:**

You can also access the REST Enabled SQL page from Workspace Utilities. On the Workspace home page, click **Workspace Utilities** and then click **REST Enabled SQL**.

2. On the REST Enabled SQL page, click **Create**.
3. On General, specify:
 - a. Name - Enter a descriptive name for this REST Enabled SQL service.
 - b. Endpoint URL - Enter the base URL for this REST Enabled SQL service.

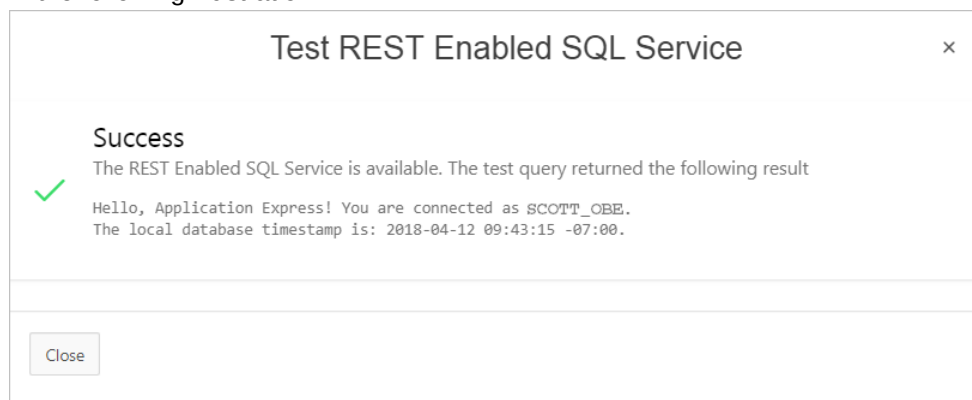
 **Tip:**

The URL must be in the format `http://host:port/ords/schema`. You obtain the URL by enabling the target schema on the remote database to be accessed by running `ORDS.ENABLE_SCHEMA`. To learn more, see "[Before You Begin: REST Enabled SQL Service Requirements](#)."

4. Click **Next**.
4. For Authentication, specify the appropriate credentials:
 - a. Credentials - Select an existing credential to authenticate against, or select **Enter new** and complete the remaining steps.
 - b. Credential Name - Enter a descriptive name for the credentials.

- c. Authentication Type - Specify the Authentication type. Select either **Basic Authentication** or **OAuth2 Client**.
 - d. Client ID:
 - For **Basic Authentication**, enter the database user name. User names for Basic Authentication are converted to upper case. To include a case-sensitive user name, prefix it with the exclamation mark (!).
 - For **OAuth2**, enter the client ID. This information will not be encrypted.
 - e. Client Secret - Enter the password, or OAuth2 Client Secret. This information is stored encrypted and cannot be retrieved in clear text.
 - f. Verify Client Secret - Enter the password or client secret again to verify your input. When the two values are not identical, an error message displays.
5. Click **Create**.

Oracle Application Express tests the REST Enabled SQL service. If everything has been done correctly (that is, you have included the correct URL, schema name, and password), the Test REST Enabled SQL Service message appears as shown in the following illustration.



The REST Enabled SQL service appears on the REST Enabled SQL page.

Example: Creating a REST Enabled SQL Service Reference

Review an example of creating REST Enabled SQL service reference on a remote database.

The following example builds a fictional REST Enabled SQL service reference named SCOTT_OBE Example using the Base URL `http://server.example.com:8080/ords/scott_obe`.

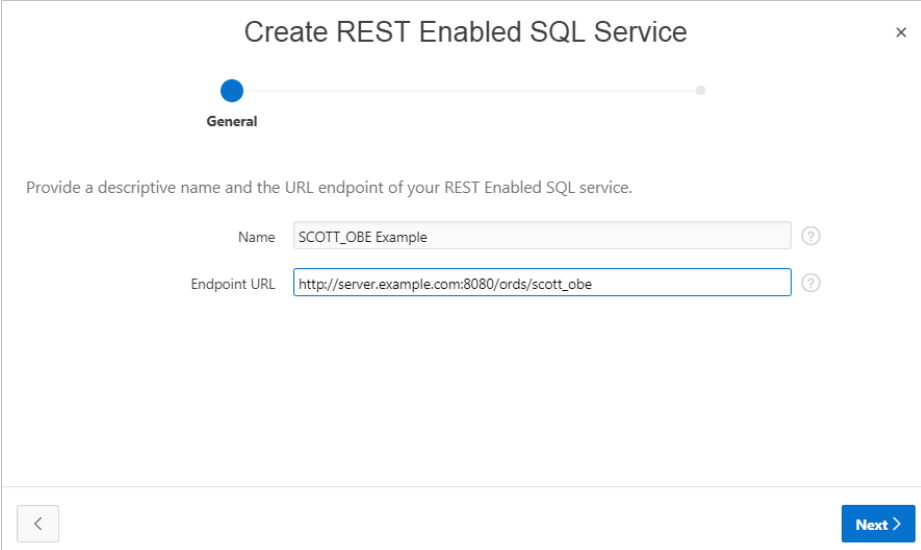
Tip:

Before creating a REST enabled SQL reference, developers must complete the requirements described in "[Before You Begin: REST Enabled SQL Service Requirements](#)".

To create a REST Enabled SQL service reference:

1. Navigate to the REST Enabled SQL page:

- a. On the Workspace home page, click **App Builder**.
 - b. Select an existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources (bottom left of the page) select **REST Enabled SQL**.
2. On the REST Enabled SQL page, click **Create**.
 3. On General, specify:
 - a. Name - Enter a descriptive name for this REST Enabled SQL service. For example: SCOTT_OBE Example.
 - b. Endpoint URL - Enter the base URL for this REST Enabled SQL service. For example: `http://server.example.com:8080/ords/scott_obe`



The screenshot shows a dialog box titled "Create REST Enabled SQL Service" with a close button (X) in the top right corner. A progress bar at the top indicates the "General" step is active. Below the title, there is a sub-header "General" and a descriptive text: "Provide a descriptive name and the URL endpoint of your REST Enabled SQL service." There are two input fields: "Name" with the value "SCOTT_OBE Example" and "Endpoint URL" with the value "http://server.example.com:8080/ords/scott_obe". Both fields have a question mark icon to the right. At the bottom left, there is a back arrow button, and at the bottom right, there is a blue "Next >" button.

- c. Click **Next**.
4. For Authentication, specify the appropriate credentials:
 - a. Credentials - Select an existing credential to authenticate against. For example: To create new credentials, select **Enter New**.
 - b. Credential Name - Enter a descriptive name for the credentials. For example: SCOTT_OBE Credentials.
 - c. Authentication Type - Specify the Authentication type. For example: **Basic Authentication**.
 - d. Client ID - Enter the database user name. For example: SCOTT_OBE.
User names for Basic Authentication are converted to upper case. To include a case-sensitive user name, prefix it with the exclamation mark (!).
 - e. Client Secret - Enter the database user name.
 - f. Verify Client Secret - Enter the password.

Create REST Enabled SQL Service - Authentication

Credentials: - Enter new -

Credential Name: SCOTT_OBE Credentials

Authentication Type: Basic Authentication

Warning: Oracle recommends the use of OAuth Client Credentials for authentication with REST enabled SQL.

* Client ID: SCOTT_OBE

* Client Secret: *****

* Verify Client Secret: ****

Create

5. Click **Create**.

Oracle Application Express tests the REST Enabled SQL service. If everything has been done correctly (that is, you have included the correct URL, schema name, and password), the Test REST Enabled SQL Service message appears as shown in the following illustration.

Test REST Enabled SQL Service

Success

The REST Enabled SQL Service is available. The test query returned the following result

Hello, Application Express! You are connected as SCOTT_OBE.
The local database timestamp is: 2018-04-12 09:43:15 -07:00.

Close

The REST Enabled SQL service appears on the REST Enabled SQL page.

Editing a REST Enabled SQL Service Reference

Edit existing REST Enabled SQL references on the Edit page.

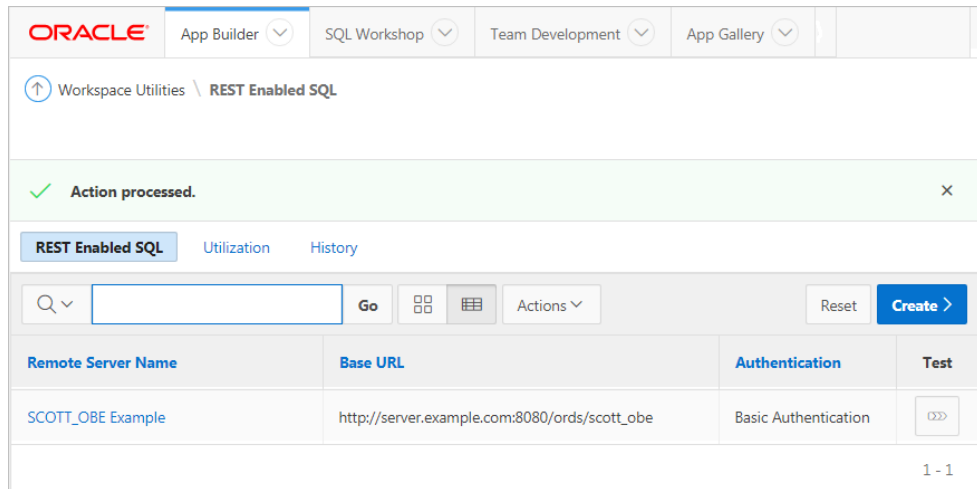
To edit a REST Enabled SQL service:

1. Navigate to the REST Enabled SQL page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **REST Enabled SQL**.

 **Tip:**

You can also access the REST Enabled SQL page from Workspace Utilities. On the Workspace home page, click **Workspace Utilities** and then click **REST Enabled SQL**.

The REST Enabled SQL page appears listing the Remote Server Name, Base URL, and Authentication.



| Remote Server Name | Base URL | Authentication | Test |
|--------------------|---|----------------------|-------------------------------------|
| SCOTT_OBE Example | http://server.example.com:8080/ords/scott_obe | Basic Authentication | <input type="button" value="Test"/> |

2. Click the remote server name.
The Edit page appears. Edit the attributes.
3. REST Enabled SQL attributes:
 - a. Name - Enter a descriptive name for this REST Enabled SQL service.
 - b. Endpoint URL - Enter the base URL for this REST Enabled SQL service.
 - c. Prompt on Install - Choose whether prompts for this REST Enabled SQL service should display when the application is being imported on another Application Express instance.
 - d. Authentication Required - Choose whether this REST Enabled SQL service requires authentication.
 - e. Credential - Select the credential to authenticate against.

 **Tip:**

To edit credentials, go to **Shared Components, Credentials**.

4. Session attributes:
 - a. Initialization Code - Enter code to be executed immediately after connecting to the REST Enabled SQL service and before the component SQL is being executed.
 - b. Cleanup Code - Enter code to be executed immediately after the component SQL is being executed.

5. Advanced attributes:
 - a. Static ID - Use the Static ID to reference the Remote Server in API Calls. Static IDs are also used to identify an existing Remote Server when the application is being exported and imported to another workspace.
 - b. Server Time Zone - Time Zone which the REST Enabled SQL service uses to decode DATE and TIMESTAMP values. This value is updated each time you click **Save and Test**.
6. Click **Apply Changes** or **Save and Test**.



See Also:

["Managing Web Credentials"](#)

Deleting a REST Enabled SQL Service Reference

Delete REST Enabled SQL references on the Edit page by clicking **Delete**.

You cannot delete a REST Enabled SQL reference if it is being used by an Application Express component.

To edit a REST Enabled SQL service:

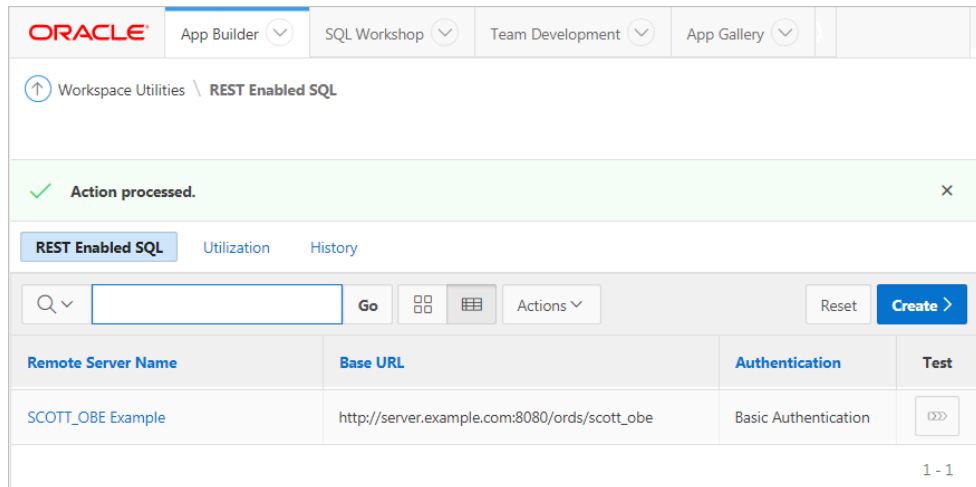
1. Navigate to the REST Enabled SQL page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **REST Enabled SQL**.



Tip:

You can also access the REST Enabled SQL page from Workspace Utilities. On the Workspace home page, click **Workspace Utilities** and then click **REST Enabled SQL**.

The REST Enabled SQL page appears listing the Remote Server Name, Base URL, and Authentication.



2. Click the remote server name.
The Edit page appears.
3. Click **Delete**.

Creating or Updating Components to Use a REST Enabled SQL

Create or update classic reports, interactive reports, CSS calendars, JET charts, Tree regions, Toggle Column reports, and Reflow Table reports to point to data on a remote database using a REST Enabled SQL reference.

- [Creating an Interactive or Classic Report Using a REST Enabled SQL Reference](#)
Create a classic or interactive report on a remote database using a REST Enabled SQL reference.
- [Creating a Chart Using a REST Enabled SQL Reference](#)
Create a chart on a remote database using a REST Enabled SQL reference.
- [Creating a Calendar Using a REST Enabled SQL Reference](#)
Create a calendar on a remote database using a REST Enabled SQL reference.
- [Specifying a REST Enabled SQL Reference in an Existing Component](#)
Configure existing classic reports, interactive reports, CSS calendars, JET charts, Tree regions, Toggle Column reports, and Reflow Table reports to point to a remote database using a REST Enabled SQL service reference.

Creating an Interactive or Classic Report Using a REST Enabled SQL Reference

Create a classic or interactive report on a remote database using a REST Enabled SQL reference.

The following example demonstrates how to create interactive report based on a table using a REST Enabled SQL reference. The procedure for creating a classic report is similar.

 **Tip:**

Before creating a report using a REST Enabled SQL reference, you must complete the tasks described in "[Before You Begin: REST Enabled SQL Service Requirements](#)" and create the reference as described in "[Creating a REST Enabled SQL Service Reference](#)."

To create an interactive report using REST Enabled SQL reference:

1. Create a REST Enabled SQL reference.
2. On the Workspace home page, click the **App Builder** icon.
3. Select the application.
4. Click **Create Page**.
5. On Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Report**.
6. On Create Page, select **Interactive Report**.

 **Tip:**

To create a Classic report, select **Classic Report** instead.

7. For Page Attributes:
 - a. Page Number - Enter a page number. If you identify a new page number, the wizard creates a new page. If you identify an existing page number, the wizard adds the component to that page.
 - b. Page Name - Specify a name for the page.
 - c. Page Mode - Identify the page mode.
 - d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - e. Click **Next**.
8. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
9. On Report Source:
 - a. Data Source - Select **REST Enabled SQL Service**.
 - b. REST Enabled SQL Service - Select a previously defined reference.
 - c. Source Type - Specify the source of the new page. Reports can be based on tables or SQL queries. Select either **Table** or **SQL Query**. In the illustration that follows, the Source Type is **Table**.

- d. Table/View Owner - Select the owner of the table on which you are building a report.
- e. Table/View Name - Select the table on which you are building a report.
- f. Select Columns - Select one or more columns to be included in the region. Selected columns will display in the report.

For example, the next illustration shows a Report Source which uses a REST Enabled SQL Service called **Example REST Enabled SQL Service Reference**, which includes all columns in the EMP table.

The screenshot shows the 'Create Interactive Report' dialog box. At the top, there are three progress indicators: a green checkmark, another green checkmark, and a blue circle labeled 'Report Source'. Below the progress indicators, the following fields are visible:

- Data Source: REST Enabled SQL Service (dropdown)
- REST Enabled SQL Service: Example REST Enabled SQL Reference (dropdown)
- Source Type: Table SQL Query (with a help icon)
- Table / View Owner: Current Schema (dropdown with a help icon)
- Table / View Name: EMP (text input with an up arrow and a help icon)
- Select Columns (with a help icon): A list of columns with arrows for selection: EMPNO (Number), ENAME (Varchar2), JOB (Varchar2), MGR (Number), HIREDATE (Date), SAL (Number), COMM (Number), and DEPTNO (Number).

At the bottom, there are 'Cancel' and 'Create' buttons.

10. Click **Create**.
Page Designer appears.
11. Click **Save and Run Page** to view the report.

Creating a Chart Using a REST Enabled SQL Reference

Create a chart on a remote database using a REST Enabled SQL reference.



Tip:

Before creating a report using a using a REST Enabled SQL reference, you must complete the tasks described in "[Before You Begin: REST Enabled SQL Service Requirements](#)" and create the reference as described in "[Creating a REST Enabled SQL Service Reference](#)."

To create a chart using a REST Enabled SQL reference:

1. Create the REST Enabled SQL reference.

2. On the Workspace home page, click the **App Builder** icon.
3. Select the application.
4. Click **Create Page**.
5. On Create a Page:
 - a. User Interface - Select a user interface for the page (optional).

This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Chart**.
6. For Chart Type, select a chart type (for example, **Pie**) and click **Next**.
7. For Page and Region Attributes:
 - a. Page Number - Select a page in which the chart object is to appear.
 - b. Page Name - Specify a name for the page.
 - c. Page Mode - Identify the page mode.
 - d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - e. Click **Next**.
8. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
9. On Source:
 - a. Location - Select **REST Enabled SQL Service**.
 - b. REST Enabled SQL Service - Select a previously defined reference.
 - c. Source Type - Specify the source of the new page. Select either **Table** or **SQL Query**.

The UI changes based on your selection. In the illustration that follows, the Source Type is **Table**. If you select, **SQL Query** follow the on-screen instructions.
 - d. Table/View Owner - Select the owner of the table on which you are building the chart.
 - e. Table/View Name - Select the table on which you are building the chart.
 - f. Page Items to Submit - (Optional) Enter a comma separated list of page items on the current page to be set into session state when the chart data gets read with a separate request.
 - g. Maximum Rows - (Optional) Enter the maximum number of rows you want to use to display the chart.

For example, the next illustration shows a Source page uses a REST Enabled SQL Service reference called **Example REST Enabled SQL Service Reference** to define a chart on the `EMP` table.

The screenshot shows the 'Create Chart' dialog box. At the top, there is a progress bar with four steps. The first three steps are completed, indicated by green checkmarks. The fourth step is currently active, indicated by a blue circle and the label 'Source'. Below the progress bar, the following fields are visible:

- Location: REST Enabled SQL Service
- REST Enabled SQL Service: Example REST Enabled SQL Reference
- Source Type: Table SQL Query
- * Table / View Owner: Current Schema
- * Table / View Name: EMP
- Page Items to Submit: [empty]
- Maximum Rows: [empty]

At the bottom of the dialog, there are three buttons: a back arrow, a 'Cancel' button, and a 'Next >' button.

h. Click **Next**.

10. On Column Mapping:

 **Tip:**

The UI that appears depends upon the selected chart type.

- a. Label Column - Select the column name to be used for defining the label(s) of the x-axis on the chart.
- b. Value Aggregation - (Optional) Select how to aggregate the chart Value Column.
- c. Value Column - Select the column name to be used for defining the value on this chart.

For example, the next illustration shows a Column Mapping based on the Label Column, `JOB`, and the Value Column, `SAL`.

Create Chart ×

Column Mapping

Chart Type: **Pie** ?

* Label Column: ?

Value Aggregation: ?

* Value Column: ?

11. Click **Create**.
Page Designer appears.
12. Click **Save and Run Page** to view the chart.

See Also:
["Creating Charts"](#)

Creating a Calendar Using a REST Enabled SQL Reference

Create a calendar on a remote database using a REST Enabled SQL reference.

Tip:

Before creating a report using a using a REST Enabled SQL reference, you must complete the tasks described in "[Before You Begin: REST Enabled SQL Service Requirements](#)" and create the reference as described in "[Creating a REST Enabled SQL Service Reference.](#)"

To create an calendar REST Enabled SQL Reference:

1. Create REST Enabled SQL reference.
2. On the Workspace home page, click the **App Builder** icon.
3. Select the application.
4. Click **Create Page**.

5. On Create a Page:
 - a. User Interface - Select a user interface for the page (optional).

This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Component** and then **Calendar**.
6. For Page Attributes:
 - a. Page Number - Enter a page number. If you identify a new page number, the wizard creates a new page. If you identify an existing page number, the wizard adds the component to that page.
 - b. Page Name - Specify a name for the page.
 - c. Page Mode - Identify the page mode.
 - d. Page Group - Identify the name of the page group you would like to associate with this page.
 - e. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - f. Click **Next**.
7. For Navigation Menu:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
8. On Source:
 - a. Data Source - Select **REST Enabled SQL Service**.
 - b. REST Enabled SQL Service - Select a previously defined reference.
 - c. Source Type - Specify the source of the new page. Reports can be based on tables or SQL queries. Select either **Table** or **SQL Query**. In the illustration that follows, the Source Type is **Table**.
 - d. Table/View Owner - Select the owner of the table on which you are building a report.
 - e. Table/View Name - Select the table on which you are building a report.
 - f. Select Columns - Select one or more columns to be included in the region. Selected columns will display in the report.
9. For Settings:
 - a. Display Column - Select the column which holds the text displayed for events on this calendar.
 - b. Start Column - Select the column which holds the start date for events displayed on this calendar.
 - c. End Date Column - Select the column which holds the end date for events displayed on this calendar. If this attribute is specified, then the calendar displays duration based events
 - d. Show Time - Select whether the time portion of the date should be displayed. The Week and Day views only be display when Show Time is set to **Yes**. If the start date or end date columns do not include time components they will be shown as 12:00 am.

For example, the next illustration shows Settings in which the Display Column is Name, the Label Column, START_DATE, and the End Date Column is END_DATE.

The screenshot shows the 'Create Page' wizard in the 'Settings' step. The progress bar at the top indicates that the first four steps are completed (green checkmarks) and the fifth step, 'Settings', is the current step (blue circle). Below the progress bar, there are four dropdown menus with help icons (question marks):

- Display Column: NAME
- Start Date Column: START_DATE
- End Date Column: END_DATE
- Show Time: Yes

At the bottom of the wizard, there are three buttons: a back arrow, a 'Cancel' button, and a blue 'Create' button.

- e. Click **Create**.

Page Designer appears.

10. Click **Save and Run Page** to view the report.

See Also:

["Understanding Page Types in the Create Page Wizard"](#)

Specifying a REST Enabled SQL Reference in an Existing Component

Configure existing classic reports, interactive reports, CSS calendars, JET charts, Tree regions, Toggle Column reports, and Reflow Table reports to point to a remote database using a REST Enabled SQL service reference.

Before configuring a component to use a REST Enabled SQL service reference, you must complete the tasks described in ["Before You Begin: REST Enabled SQL Service Requirements"](#) and create the reference as described in ["Creating a REST Enabled SQL Service Reference."](#)

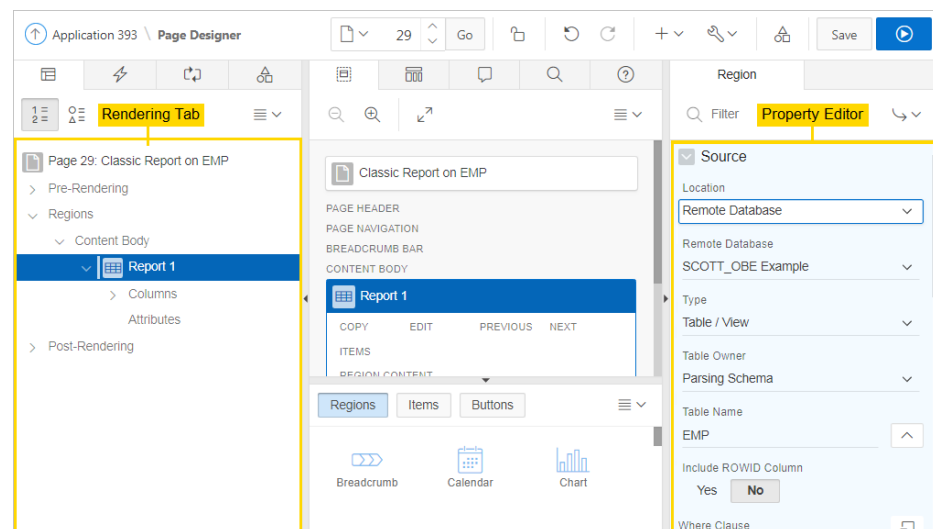
To specify a remote database in an existing component:

1. Create the REST Enabled SQL reference.
2. View the page containing the component in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

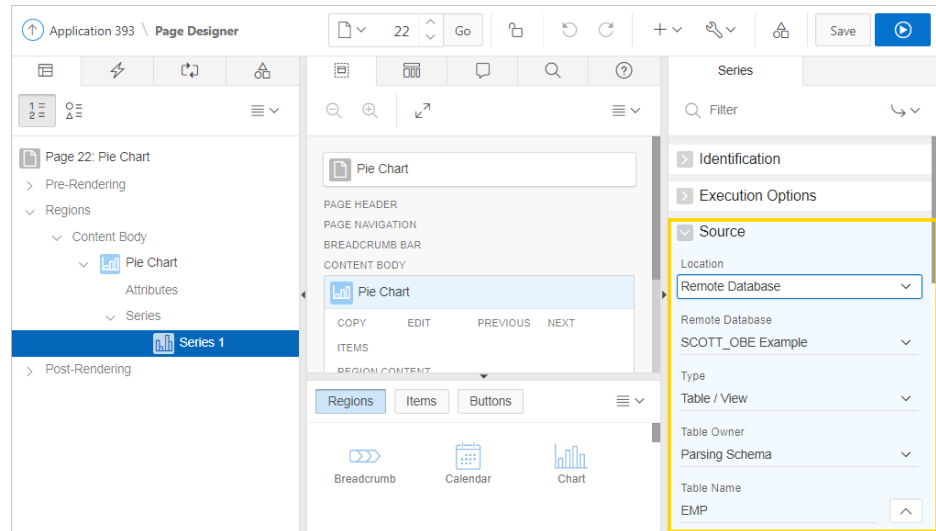
3. In Page Designer, select the region containing the component (if not already selected) and edit the Source attributes.
4. To specify a remote database for an interactive report, a classic report, or CSS calendar:
 - a. In the Rendering tab (left pane), select the region containing the report or calendar.
 - b. In the Property Editor (right pane), find the **Source** group.
 - c. Edit the following Source attributes:
 - Location - Select **Remote Database**.
 - Remote Database - Select the defined REST Enabled SQL Service reference.
 - Type - Select how the data is queried. Select **Table/View** or **SQL Query**.
 - Configure the table or provide a SQL Query.
 - Configure the remaining attributes as needed.

In the following example, the region **Report 1** is selected and the Source points to a Remote Database called **SCOTT_OBE Example**.



5. To specify a remote database for chart:
 - a. In the Rendering tab (left pane), find and expand the region containing the chart.
 - b. Expand **Series** and select the Series that defines the chart.
 - c. In the Property Editor (right pane), find the **Source** group.
 - d. Edit the following Source attributes:
 - Location - Select **Remote Database**.
 - Remote Database - Select the defined REST Enabled SQL Service reference.
 - Type - Select how the data is queried. Select **Table/View** or **SQL Query**.
 - Configure the table or provide a SQL Query.
 - Configure the remaining attributes as needed.

In the following example, **Series 1** is selected and the Source points to a Remote Database called **SCOTT_OBE Example**.



6. Click **Save** or **Save and Run Page**.

Managing Web Source Modules

Web Source Modules act as a reference to one or multiple external web services. A module can contain one or many Web Source Operations which are the references to a concrete external web service.

- [About Web Source Modules](#)
- [Creating a Web Source Module](#)
- [Editing or Deleting a Web Source Module](#)
- [Copying a Web Source Module](#)
- [Viewing Credential Utilization](#)
- [Viewing Web Service Module History](#)

About Web Source Modules

How Web Sources Differ from Web Service References

Web Source Modules enable developers to access to Representational State Transfer (REST) services or generic JSON data feeds in applications applications and use the data in Application Express components such as reports, interactive reports, and interactive grids. Unlike existing Web Service References, a Web Source Module contains metadata about the Web service which can be used by Application Express Components or PL/SQL processes to invoke the service and to process the responses.

Web Source Modules contain multiple operators that differ depending upon the Web service target. For a REST services, an operation is a specific service handler (such as, GET, PUT, POST, or DELETE). Developers assign Operation a Database Action such as Fetch Multiple Rows, Fetch Single Row, Insert Row, Update Row, and Delete

Row. However, you can assign each Database Operation only once to a Web Source Operation.

Integration with Application Express Components

Oracle Application Express provides direct integration of Web Source Modules classic reports, interactive reports, CSS Calendar, and JET Charts.

About Remote Servers

Oracle Application Express splits the endpoint URL of a Web Service into two parts. The server-specific part is stored as a separate entity called the **Remote Server**. You can reuse a Remote Server with multiple Web Source Modules if it uses the same server, port and URL Path Prefix (context root).

If you change the attributes of a Remote Server, the change impacts all Web Source Modules using the Remote Server. Remote Servers make it easy to move a collection of Web Source Modules. For example, you can move from test system to a production system by changing the URL within the Remote Server object.

About Authentication and Credentials

Web Source Modules support both Basic Authentication and the OAuth Client Credentials flow. Authentication credentials can be specified at the Remote Server-level for all Web Source Modules using the Remote Server). If credentials are set at the Web Source Module-level, that setting supersedes credentials stored at the Remote Server level.

A Credential denotes the Authentication method, a Client ID (or user name) and a Client Secret (or password). Credentials are stored as a named entity within Shared Components and can be re-used across multiple Web Source Modules or Remote Servers.



See Also:

["Managing Remote Servers"](#) and ["Managing Web Credentials"](#)

Creating a Web Source Module

To create a Web Source Module:

1. Navigate to the Web Service Modules page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Web Source Modules**.
2. On the Web Source Modules page, click **Create**.
3. For Create Web Source Module - Method, select **From Scratch** and click **Next**.

 **Tip:**

If you select, **As copy of an existing web source module** select the application to copy from and follow the on-screen instructions.

4. Specify the following attributes:

- a.** Web Source Type - Select a web source type.

If the REST service you are about to access is provided by Oracle REST Data Services (ORDS), select **Oracle REST Data Services**. Otherwise, select **Simple HTTP**. To learn more, see field-level Help.

- b.** Name - Enter a descriptive name for this Web Source Module.

- c.** Endpoint URL - Enter the Endpoint URL (starting with `http://` or `https://`) for this web source.

- d.** HTTPS Host Name - This attribute only displays if your configuration is running on Oracle Database 12.2 or higher.

Enter the The host name to be matched against the common name (CN) of the remote server's certificate for an HTTPS request. For example, enter a domain name such as `*.example.com`. If NULL, the host name in the given URL is used.

- e.** HTTP Method - Choose the HTTP Method to use for Service Discovery.

- f.** Click **Next**.

Application Express tests the URL endpoint once you click **Next**. If the URL is incorrect, an error message displays. If that is the case, click **Back** and correct the URL. If the URL works, Remote Server appears.

5. For Remote Server: **Tip:**

To learn more about an attribute, see field-level Help.

- a.** Remote Server - Choose an existing Remote Server object. The HTTP Endpoint URL of a Web Source Module consists of the remote server (or Base URL) and the service specific part.

- b.** Base URL - Base URL of the chosen Remote Server object.

- c.** Service URL Path - Enter the service-specific URL Path for the Web Source Module. The HTTP Endpoint URL is built by appending this to the Base URL specified in the Remote Server object.

For example:

`/services/customers/`

`/products/4711/details`

- d.** Click **Next**.

6. Authentication Required - Choose whether the new Web Source Module requires authentication. Select **Yes** or **No**. If you select **Yes**, follow the on-screen instructions.
7. To proceed, select one of the following buttons:
 - **Create Module Manually** - This option does not include Web Source Discovery mode. Based on the previously entered information, a Web Source Module is created with a few sample operations and a sample data profile. **Create Module Manually** creates an *incomplete* Web Source Module that is unusable with an Oracle Application Express component. To correctly reflect the external web service, you will need to edit the web source operations and the data profile.
 - **Discover** - Starts Web Source Discovery mode. Oracle Application Express executes multiple HTTP requests in order to get information about the external Web service. Application Express samples the data (JSON or XML) returned by the Web service and derives a Data Profile (consisting of metadata about the JSON attributes). Discovery results then display and confirms whether to create the Web Source Module.
 - **Advanced** - Presents advanced parameters for the Data Profile including HTTP Headers, URL Parameters, and additional details.
8. If you click **Advanced**, the Parameter page appears. **Advanced** enables you to configure some initial Web source parameters which might be required for discovery. Some Web sources may require you set specific HTTP Headers (for example, *User-Agent*). After setting advanced parameters, select either **Discover** or **Create Module Manually** buttons.
9. If you select **Discover**, the Web Source Discovery Preview page appears.

The information that displays depends upon the complexity of the target service. For simple HTTP services or ORDS services which do not provide metadata about their JSON response attributes, Application Express investigates the JSON or XML response, samples the data, and creates a proposal for a data profile. The Data tab displays the sample response data, formatted as a report.

The columns tab shows information about the data profile columns Application Express has found during discovery. Use this tab to verify whether the column data types are correct.

Click **More Detail** to view additional tabs that show more detailed information about the Web Source response, such as the response body, response headers and the Web Source operations.
10. Click **Create Web Source**.

Editing or Deleting a Web Source Module

To edit a Web Source Module:

1. Navigate to the Web Service Modules page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Web Source Modules**.

2. On the Web Source Modules page, click **Module Name**.
3. To delete the Web Service Module, click **Delete**. Otherwise, edit the attributes.
4. Under Web Source Module:
 - a. Name - Enter a descriptive name for this Web Source Module.
 - b. Web Source Type - Options include:
 - **Simple HTTP** - Select this option for a simple HTTP data feed. Application Express assumes that all data is returned with the first request and that the server does not support server-side filtering, ordering or other advanced REST service feature. All invocation details and parameters must be configured manually.
 - **Oracle REST Data Services** - Select this option for Oracle REST Data Services (ORDS) REST Services. These REST services follow the Oracle REST standard, which means that `GET`, `POST`, `PUT` or `DELETE` operations are standardized. Server-side filtering and ordering are supported. Application Express can leverage these features by delegating report
 - **Oracle REST Data Services (Legacy Syntax)** - Select this option for services powered by older versions of ORDS (Oracle REST Data Services). These services are not completely compliant to the Oracle REST Standard.
 - **ORDS (Legacy)** - Select this option for services powered by older versions of Oracle REST Data Services (ORDS). These services are not completely compliant to the Oracle REST Standard.
 - c. Remote Server - Indicates the remote server on which the Web Source is hosted.
 - d. HTTPS Host Name - Enter the The host name to be matched against the common name (CN) of the remote server's certificate for an HTTPS request. For example, the HTTPS Host Name can be a domain name like `*.example.com`. If NULL, the host name in the given URL is used.
 - e. Base URL - Lists the Base URL.
 - f. URL Prefix - URL Path Prefix pointing to this Web Source Module. This prefix is being appended to the Remote Server Base URL.
5. Under Data Profile:
 - a. Click **Edit Data Profile** to change how Web source responses (in XML or JSON) format are being parsed and converted to rows and columns.
The Data Profile page appears.
 - b. Under **Data Profile** and **Columns**, edit the attributes.
Row Selector stores an XML or JSON path expression pointing to the node containing the collection of rows. **Columns** determine how one row is parsed and converted to multiple columns.
 - c. Click **Apply Change** to save your changes.
6. **Subscription** lists Web Source Modules that subscribe to this module.
7. Under Operations:
 - a. Operation - Click **Add Operation** to add a Web Source Operation. A web source **Operation** contains the actual information about the external web service handler.

1. Navigate to the Web Service Modules page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Web Source Modules**.
2. On the Web Source Modules page, click **Utilization**.

The Utilization page displays where a Web Source Module is used within the application.

Viewing Web Service Module History

To view recent modifications to Web Source Modules:

1. Navigate to the Web Service Modules page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Web Source Modules**.
2. On the Web Source Modules page, click **History**.

The History page displays recent modifications made to Web Service Modules in the current workspace.

Managing Remote Servers

A Remote Server is separate entity that stores Web Source server information. Remote Servers can be shared among multiple Web Sources, thus enabling you to information such as the Base URL or Authentication.

- [About Remote Servers](#)
- [Creating a Remote Server](#)
- [Editing or Deleting a Remote Server](#)
- [Viewing Remote Server Utilization](#)
- [Viewing Remote Server History](#)

About Remote Servers

Oracle Application Express splits the endpoint URL of a Web Source Module into two parts. The first part is server-specific part and is stored as a separate entity called the **Remote Server**. You can reuse a Remote Server with multiple Web Source Modules if each one uses the same server, port, and URL Path Prefix (context root). Remote Servers are stored at the workspace-level and therefore visible in all applications.

The second part of the endpoint URL and is specific to the Web Source module. Multiple Web Source modules can share one Remote Server, thus sharing information such as the Base URL and Authentication. If you change Remote Server attributes,

the change impacts all Web Source Modules using the Remote Server. Remote Servers make it easy to move a collection of Web Source Modules. For example, you can move from a test system to a production system by changing the URL within the Remote Server object.

Exporting and Importing Remote Server Information

When you export an application, referenced Remote Servers are added to the export file. When you import the application into another workspace, Application Express checks whether the target workspace already contains Remote Servers with the same **static ID**. If a Remote Server already exists, the application uses it. Otherwise the Remote Servers from the import file are created in the target workspace.



See Also:

["Managing Web Source Modules"](#)

Creating a Remote Server

To create a Remote Server object:

1. Navigate to the Remote Server page:
 - From Workspace Utilities:
 - a. On the Workspace home page, click **App Builder**.
 - b. Click **Workspace Utilities**.
 - c. Click **Remote Server**.
 - From Shared Components:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Workspace Objects, select **Remote Servers**.
2. On the Remote Servers page, click **Create**.
3. Name - Enter a name for the remote server object. Remote Server names must be unique within the application.
4. Static ID - Enter a static ID for the Remote Server.
5. Server Type - Select one of the following:
 - **Web Source**
 - **Authentication**
6. Base URL - Enter the base URL of the remote server.
7. Prompt on Install - Select Yes to have Oracle Application Express prompt for the Base URL after the application installs into another workspace.

8. Comments - Enter any comments or notes here. These comments never display when running the application.
9. Click **Create**.

Editing or Deleting a Remote Server

To edit a Remote Server:

1. Navigate to the Credentials page:
 - From Workspace Utilities:
 - a. On the Workspace home page, click **App Builder**.
 - b. Click **Workspace Utilities**.
 - c. Click **Remote Server**.
 - From Shared Components:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Security, select **Remote Server**.
2. On the Remote Server page, click the name of the Remote Server.
The Edit Remote Server dialog appears.
3. To delete the current Remote Server, click **Delete**. Otherwise, edit the appropriate attributes.

 **Tip:**

Remote Servers which are referenced within an application cannot be deleted.

4. To edit the attributes:
 - a. Name - Enter a descriptive name.
 - b. Static ID - Enter a name for the remote server object here. Remote Server names must be unique within the application.
 - c. Server Type - Select either:
 - **Web Source**
 - **Authentication**
 - d. Base URL - Enter the base URL of the remote server.
 - e. Prompt on Install - If **Yes**, Application Express prompts for the Base URL after the application has been installed into another workspace.
 - f. HTTPS Host Name - The host name to be matched against the common name (CN) of the remote server's certificate for an HTTPS request. It can also be a domain name like *.example.com. If NULL, the host name in the given URL is used.

HTTPS Host Name only displays if you are using Oracle Database 12.2 or higher.

- g.** Comments - Enter any comments or notes. These comments only display within App Builder and never display when running the application.
- h.** To save your changes, click **Apply Changes**.

Viewing Remote Server Utilization

To view Web Source Modules utilization:

- 1.** Navigate to the Remote Server page:
 - From Workspace Utilities:
 - a.** On the Workspace home page, click **App Builder**.
 - b.** Click **Workspace Utilities**.
 - c.** Click **Remote Server**.
 - From Shared Components:
 - a.** On the Workspace home page, click **App Builder**.
 - b.** Select an application.
 - c.** On the Application home page, click **Shared Components** in the center of the page.
 - d.** Under Workspace Objects, select **Remote Servers**.
- 2.** On the Remote Servers page, click **Utilization**.

The Utilization page displays where Remote Servers page used within the current workspace.

Viewing Remote Server History

To view recent modifications to Remote Servers:

- 1.** Navigate to the Remote Server page:
 - From Workspace Utilities:
 - a.** On the Workspace home page, click **App Builder**.
 - b.** Click **Workspace Utilities**.
 - c.** Click **Remote Server**.
 - From Shared Components:
 - a.** On the Workspace home page, click **App Builder**.
 - b.** Select an application.
 - c.** On the Application home page, click **Shared Components** in the center of the page.
 - d.** Under Workspace Objects, select **Remote Servers**.
- 2.** On the Remote Servers page, click **History**.

The History page displays recent modifications made to Remote Servers in the current workspace.

Managing Web Credentials

Manage secure credentials to connect to REST Enabled SQL or other REST services.

- [About Credentials](#)
- [Creating Web Credentials](#)
- [Editing or Deleting Credentials](#)
- [Viewing Credential Utilization](#)
- [Viewing Credential History](#)

About Credentials

Use **Credentials** to connect to REST Enabled SQL or other external REST services. Oracle Application Express securely stores and encrypts these credentials for use by Application Express components. Credentials cannot be retrieved back in clear text. Credentials are stored at the workspace-level and therefore are visible in all applications.

Exporting and Importing Credentials

When you export an application, used credentials are added to the export file. When you import the application into another workspace, Application Express checks whether the target workspace already contains credentials with the same static ID. If a credential already exists, the application uses it. Otherwise the credential from the import file is created in the target workspace.



See Also:

["Managing REST Enabled SQL References"](#)

Creating Web Credentials

You can create credentials from either Workspace Utilities or Shared Components.

To create credentials:

1. Navigate to the Web Credentials page:
 - From Workspace Utilities:
 - a. On the Workspace home page, click **App Builder**.
 - b. Click **Workspace Utilities**.
 - c. Click **Web Credentials**.
 - From Shared Components:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.

- c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Security, select **Web Credentials**.
 2. On the Web Credentials page, click **Create**.
 3. Configure the Attributes on the Web Credentials page.
 4. For Name - Enter a descriptive name.
 5. Authentication Type - Select one of the following:
 - **Basic Authentication**
 - **OAuth2 Client Credentials**
 6. Client ID or User Name - Select one of the following:
 - For **Basic Authentication**, enter the user name.
 - For **OAuth2 Client Credentials**, enter the client ID.
This information is not be encrypted.
 7. Client Secret or Password - Enter the password or OAuth2 Client Secret.
This information will be stored encrypted and cannot be retrieved in clear text.
 8. Verify Client Secret or Password - Enter the password or client secret again to verify your input. When the two values are not identical, an error message displays.
 9. Prompt On Install - Choose whether prompts for this credential display when the application is imported on another Oracle Application Express instance.
 10. Comments - Enter any comments or notes here. These comments never display when running the application.
 11. Click **Create**.

Editing or Deleting Credentials

To edit credentials:

1. Navigate to the Web Credentials page:
 - From Workspace Utilities:
 - a. On the Workspace home page, click **App Builder**.
 - b. Click **Workspace Utilities**.
 - c. Click **Web Credentials**.
 - From Shared Components:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Security, select **Web Credentials**.
2. On the Web Credentials page, select the credential name.
3. To delete the credentials, click **Delete**. Otherwise, edit the appropriate attributes.

**Tip:**

A credential cannot be deleted when it is being referenced somewhere in the workspace.

4. To edit the attributes:
 - a. Name - Enter a descriptive name.
 - b. Authentication Type - Select a type:
 - **Basic Authentication**
 - **OAuth2 Client Credentials**
 - c. Client ID or User Name: For **Basic Authentication**, enter the user name. For **OAuth2**, enter the client ID. This information will not be encrypted.
 - d. Client Secret or Password - Enter the password or OAuth2 Client Secret.
This information is stored encrypted and cannot be retrieved in clear text.
 - e. Verify Client Secret or Password - Enter the password or client secret again to verify your input. When the two values are not identical, an error message displays.
 - f. Prompt on Install - Choose whether prompts for this credential display when the application is imported on another Application Express instance. Since credentials are not part of an application export file, Application Express always generates prompts after install, when the client ID or username is not empty.
 - g. Comments - Enter any comments or notes. These comments only display within App Builder and never display when running the application.
 - h. To save your changes, click **Apply Changes**.

Viewing Credential Utilization

To view Web credential utilization:

1. Navigate to the Web Credentials page:
 - From Workspace Utilities:
 - a. On the Workspace home page, click **App Builder**.
 - b. Click **Workspace Utilities**.
 - c. Click **Web Credentials**.
 - From Shared Components:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Security, select **Web Credentials**.
2. On the Web Credentials page, click **Utilization**.

The top of the Utilization page displays used credentials, the associated component type, and the component name. Unused credentials display at the bottom of the page.

3. To delete unused credentials, click **Delete Unused**.

Viewing Credential History

To view recent modifications to Web credentials:

1. Navigate to the Web Credentials page:
 - From Workspace Utilities:
 - a. On the Workspace home page, click **App Builder**.
 - b. Click **Workspace Utilities**.
 - c. Click **Web Credentials**.
 - From Shared Components:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Security, select **Web Credentials**.
2. On the Web Credentials page, click **History**.

The History page displays recent modifications made to Credentials in the current workspace.

Managing Legacy Web Services

Legacy Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment.



Note:

The SOAP 1.1 specification is a W3C note. SOAP Version 1.2 specification is a W3C recommendation.

For information about Simple Object Access Protocol (SOAP) see:

<http://www.w3.org/TR/SOAP/>

- [About Web Services](#)
- [Creating Web Service References](#)
- [About Working with SSL Enabled Web Services](#)
- [Creating Web Service References Based on a WSDL](#)
- [Creating Web Service References Manually](#)
- [Creating RESTful Web Service References](#)

- [Using the Web Service Reference Repository](#)
- [Creating an Input Form and Report on a Web Service](#)
- [Creating a Form on a Web Service](#)
- [Creating a Report on a Web Service](#)
- [How to Invoke a Web Service as a Process](#)
- [Viewing a Web Service Reference History](#)

About Web Services

In a typical Web services scenario, a business application sends a request to a service at a given URL by using the protocol over HTTP. The service receives the request, processes it, and returns a response. You can incorporate calls with external Web services in applications developed in App Builder.

Web services are typically based on Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) architectures. SOAP is a World Wide Web Consortium (W3C) standard protocol for sending and receiving requests and responses across the Internet. SOAP messages can be sent back and forth between a service provider and a service user in SOAP envelopes. RESTful Web services are resource oriented. The scope of the Web service is found in the URI and the method of the service is described by the HTTP method that is used such as GET, POST, PUT, HEAD, and DELETE.

SOAP offers two primary advantages:

- SOAP is based on XML, and therefore easy to use.
- SOAP messages are not blocked by firewalls because this protocol uses simple transport protocols, such as HTTP.

Tip:

If you run Oracle Application Express with Oracle Database 11g Release 1 (11.1), you must enable network services to use Web services.

REST offers similar advantages:

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

See Also:

["Enabling Network Services in Oracle Database 11g or Later"](#)

Creating Web Service References

To use Web services in Oracle Application Express, you create a Web service reference using a wizard. Web service references can be based on a Web Services Description Language (WSDL) document, RESTful style, or created manually by supplying information about the service.

- [Accessing the Web Service References Page](#)
- [Specifying an Application Proxy Server Address](#)

See Also:

- ["Creating Web Service References Based on a WSDL"](#)
- ["Creating Web Service References Manually"](#)
- ["Creating RESTful Web Service References"](#)

Accessing the Web Service References Page

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

To access the Web Service References page:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Data Sources, click **Legacy Web Service References**.
The Web Service References page appears.

Specifying an Application Proxy Server Address

If your environment requires a proxy server to access the Internet, you must either specify a proxy server address on the Application Attributes page, or your instance administrator must define an instance proxy, before you can create a Web service reference.

To specify a proxy address for an application:

1. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
2. Select an application.
Application home page appears.
3. Click **Edit Application Properties**.

4. Under Properties, enter the proxy server in the Proxy Server field.
5. Click **Apply Changes**.

About Working with SSL Enabled Web Services

Secure Sockets Layer (SSL) is an industry standard protocol that uses RSA public key cryptography with symmetric key cryptography to provide authentication, encryption, and data integrity.

If the Web service that you need to interact with is SSL-enabled (that is, `https` displays in the URL to the Web service), you must create a wallet and configure Oracle Application Express to use the wallet. A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL.



See Also:

"Configuring Wallet Information" in *Oracle Application Express Administration Guide*

Creating Web Service References Based on a WSDL

You can create Web service reference based on a Web Services Description Language (WSDL) document.

- [About Creating Web Service References Based on a WSDL](#)
- [Creating a Web Service Reference by Specifying a WSDL Document](#)
- [Testing a Web Service Reference Created from a WSDL](#)

About Creating Web Service References Based on a WSDL

Before you create a Web service reference based on a WSDL, you must decide how to locate the WSDL. You locate a WSDL by entering the URL to the WSDL document.

You then run a wizard which analyzes the WSDL and collects all the necessary information to create a valid SOAP message, including:

- The URL used to post the SOAP request over HTTP(S)
- A Universal Resource Identifier (URI) identifying the SOAP HTTP request
- Operations of the Web Service
- Input parameters for each operation
- Output parameters for each operation

Creating a Web Service Reference by Specifying a WSDL Document

To create a Web service by specifying a URL to a specific WSDL document:

1. Navigate to the Web Service References page.

- a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Legacy Web Service References**.
2. Click **Create**.
 3. For Web Reference Type, select **Based on WSDL** and click **Next**.
 4. If prompted to search a UDDI registry to find a WSDL, click **No** then **Next**.
 5. In WSDL Location, enter the URL to the WSDL document.
 6. For Username, enter the username required to access the WSDL.
 7. For Password, enter the password required to access the WSDL.
 8. Click **Next**.
The Web Service Details page appears.
 9. For Basic Authentication, select:
 - **No** - Select this option if authentication credentials are not sent as part of the HTTP request.
 - **Yes** - Select this option if authentication credentials are sent as part of the HTTP request. Web services that employ basic authentication usually also use the HTTPS (SSL) protocol since the username and password are sent in clear text.
 10. Click **Create Reference**.
The Web service reference is added to the Web Service References Repository.
 11. On the Create page, the following selections are available:
 - View Web Service References - Select to go to the Web Service Reference page.
 - Create Form on Web Service - Select to create a Form for this Web Service Reference.
 - Create Form & Report on Web Service - Select to create a Form and Report for this Web Service Reference.

Testing a Web Service Reference Created from a WSDL

After you have created a Web service reference, you can test it on the Test Web Service Reference page.

To test a Web service reference:

1. Navigate to the Web Service References page.
 - a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Legacy Web Service References**.

2. Select the **View Report** icon.
3. Click the **Test** icon for the Web Service reference you want to test.
The Test Web Service Reference page appears. The Web service name and URL endpoint display at the top of the page.
4. From Operation, select an operation (that is, the method to be executed).
5. Under Input Parameters, enter the appropriate values.
6. Click **Test**.
The message request and response appear at the bottom of the page under Messages.



See Also:

["Accessing the Web Service References Page"](#)

Creating Web Service References Manually

You can create a Web service reference manually.

- [About Creating Web Service References Manually](#)
- [Creating a Web Service Reference Manually](#)
- [Testing a Web Service Reference Created Manually](#)

About Creating Web Service References Manually

When you create a Web service reference manually, you supply the necessary information to create a valid SOAP request, including:

- The URL used to post the SOAP request over HTTP(S)
- A Universal Resource Identifier (URI) identifying the SOAP HTTP request
- The SOAP envelope for the request, including any item substitutions
- Optionally the name of a collection to store the response from the Web service

Creating a Web Service Reference Manually

Creating a Web service reference manually, adds it to Web Service References Repository.

To create a Web service reference manually:

1. Navigate to the Web Service References page.
 - a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Legacy Web Service References**.

2. Click **Create**.
3. Select **Manual** and click **Next**.
4. On Create Web Service Reference:
 - a. Name - Enter a name to identify the reference.
 - b. URL - Enter the URL used to post the SOAP request over HTTP. This corresponds to the soap:address location of a service port in the WSDL.
 - c. Action - Enter the intent of the SOAP HTTP request (optional). The value is a URI identifying the intent. SOAP places no restrictions on the format or specificity of the URI or requires that it is resolvable.
 - d. Proxy - Enter a proxy to override the application proxy for this service.
 - e. SOAP Version - Select **1.1** or **1.2**.
 - f. Basic Authentication - Choose whether the Web service requires authentication. Select **Yes** or **No**.
 - g. SOAP Envelope - Specify the SOAP envelope to be used for the SOAP request to the Web service..

 **Note:**

You can reference items from session state in the SOAP envelope by using #ITEM_NAME# syntax.

- h. Store Response in Collection - Enter the name of a collection to store the Web service response. The response will be stored in the CLOB001 column of the collection entered here.
- i. Click **Create**.

 **See Also:**

["Accessing the Web Service References Page"](#)

Testing a Web Service Reference Created Manually

After you have created a Web service reference, you can test it on the Test Web Service Reference page.

To test a Web service reference:

1. Navigate to the Web Service References page.
 - a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Legacy Web Service References**.

2. Select the **View Report** icon.
3. Click the **Test** icon for the Web Service reference you want to test.
The Test Web Service Reference page appears. The Web service name and URL endpoint display at the top of the page.
4. If required, enter the username and password under Basic Authentication.
5. In SOAP Envelope text area, optionally edit the SOAP request envelope.
6. Click **Test**.
The message request and response appear at the bottom of the page under Messages.

**See Also:**

["Accessing the Web Service References Page"](#)

Creating RESTful Web Service References

You can create a Representational State Transfer (REST) or RESTful web service.

- [About Creating RESTful Web Service References](#)
- [Creating a RESTful Web Service Reference](#)
- [Testing a REST Web Service Reference](#)

About Creating RESTful Web Service References

RESTful Web services are resource oriented. The scope of the Web service is found in the URI and the method of the service is described by the HTTP method that is used such as GET, POST, PUT, HEAD, and DELETE. When you create a RESTful Web service reference, you supply the necessary information about the structure of the request and response including:

- A Universal Resource Identifier (URI) identifying the RESTful request
- The HTTP method identifying the method of the Web service
- HTTP Headers, if required, that are part of the request
- The type of input expected by the Web service
- The format of the response and how to identify the response parameters

Creating a RESTful Web Service Reference

To create a RESTful Web service reference:

1. Navigate to the Web Service References page.
 - a. On the Workspace home page, click **App Builder**.
 - b. Select a existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.

- d. Under Data Sources, select **Legacy Web Service References**.
2. Click **Create**.
3. Select **REST** and click **Next**.
4. For REST Details, specify the following:
 - a. Name - Enter a name to identify the reference.
 - b. URL - Enter the URL endpoint of the Web service.
 - c. Proxy - Enter a proxy to override the application proxy for this service (optional). This setting is overridden by the proxy setting for this instance.
 - d. HTTP Method - Choose the http method used for the request to the Web service. Select **GET**, **HEAD**, **POST**, **PUT** or **DELETE**.
 - e. Basic Authentication - Select **Yes** to require HTTP Basic Authentication. Otherwise, select **No**.
 - f. REST HTTP Headers - Enter the names of the HTTP headers to send with the request.
 - g. Click **Next**.
5. For REST Input parameters, specify the following:
 - a. Name - Enter the name of the input parameter expected by the method.
 - b. Type - Select the input type.
 - c. Click **Add Parameter**.
 - d. Repeat steps a through c for each expected input.
 - e. Click **Next**.
6. For REST Output parameters, specify the following:
 - a. Output Format - Select **XML**, **Text** or **JSON** for the response format expected from the Web service.
 - b. XPath to Output Parameters (XML only) - Enter an XPath expression to the relevant part of the response.

For example:

```
/ListBucketResult/Contents
```
 - c. Response Namespace (XML only) - Enter the namespace corresponding to the Response XPath.

For example:

```
http://s3.amazonaws.com/doc/2006-03-01/
```
 - d. Parameter Delimiter (**Text** only) - Enter the character or sequence that separates parameters returned from the Web service. Use `\n` to indicate a new line and `\t` to indicate a tab character.
 - e. New Record Delimiter (**Text** only) - Enter the character or sequence that determines a new record in a text response from the Web service. Use `\n` to indicate a new line and `\t` to indicate a tab character.
 - f. Name - Enter the name of the output parameter returned by the method.
 - g. Path - Enter the path. If the response is XML, the path is an XPath expression to the node. If the response is Text, the path is a number. If the response is

JSON, the entire response is stored in the CLOB001 column of a collection you specify.

- h.** Type - Select the output type.
- i.** Click **Add Parameter**.
- j.** Repeat steps f through h for each returned output parameter.

 **Note:**

Click **Test** to send a request to a RESTful Web service and see the response. This test process helps you specify the appropriate Output parameters.

- 7.** Click **Create**.

The Create Web Service Reference Success page appears. The Web service reference is added to the Web Service References Repository.

 **See Also:**

- ["Accessing the Web Service References Page"](#)
- "Creating a Proxy Server for an Instance" in *Oracle Application Express Administration Guide*

Testing a REST Web Service Reference

After you have created a Web service reference, you can test it on the Test Web Service Reference page.

To test a Web service reference:

- 1.** Navigate to the Web Service References page.
 - a.** On the Workspace home page, click **App Builder**.
 - b.** Select a existing application.
 - c.** On the Application home page, click **Shared Components** in the center of the page.
 - d.** Under Data Sources, select **Legacy Web Service References**.

- 2.** Select the **View Report** icon.

- 3.** Click the **Test** icon for the Web Service reference you want to test.

The Test Web Service Reference page appears. The Web service name and URL endpoint display at the top of the page.

- 4.** In the URL and Proxy Override fields, optionally edit the values for the test.
- 5.** If required, enter the username and password under Basic Authentication.
- 6.** Under HTTP Headers, enter appropriate values.

7. Under Input Parameters, enter appropriate values.
8. Click **Test**.

The message response appears at the bottom of the page under Messages.



See Also:

["Accessing the Web Service References Page"](#)

Using the Web Service Reference Repository

Web service references are stored in the Web Service Reference Repository.

To access the Web Service References Repository:

1. Navigate to the Web Service References page.
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an existing application.
 - c. On the Application home page, click **Shared Components** in the center of the page.
 - d. Under Data Sources, select **Legacy Web Service References**.

The Web Service Reference page appears.

A Search bar displays at the top of the page. Available controls include:

- **Search columns icon** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
 - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
 - **Go button** - Executes a search or applies a filter.
 - **View Icons** - Displays each Web service reference as a large icon. To edit a Web service reference, click the appropriate icon.
 - **View Report** - Displays each Web service reference as a line in a report.
 - **Actions menu** - Displays the Actions menu. Use this menu to customize the report view.
2. Click the **View Report** icon.
 3. In report view you can:
 - Edit a reference by clicking the reference name.
 - Test a reference by clicking the **Test** icon.
 - View details about a reference by clicking the **View** icon. Note that this option is not available for manually created or REST Web service references.

 **See Also:**

- ["Accessing the Web Service References Page"](#)
- ["About the Actions Menu"](#)

Creating an Input Form and Report on a Web Service

This section describes how to create an input form and report on a Web service.

- [About the Create Form and Report on Web Service Wizard](#)
- [Creating a Form and Report by Adding a New Page](#)

About the Create Form and Report on Web Service Wizard

The Create Form and Report on Web Service Wizard creates an input form, a submit button, and a report for displaying results. You can execute this wizard directly after creating the Web service reference from a WSDL or a RESTful style Web service, or by adding a page.

Use this wizard when you expect a nonscalar result from the Web service. The Amazon Product API Web service is a good example. This Web service returns many results based on the search criteria entered in an input form.

Creating a Form and Report by Adding a New Page

If you have an existing Web service reference, you can create an input form and report by adding a new page.

To create a form and report by adding a new page:

1. Create the Web service reference.
2. Run the Create Page Wizard. On Create a Page:
 - a. For User Interface, if applicable select a user interface for the page.
This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.
 - b. For Page Type, select **Component** and then **Form**.
 - c. On Create page, select **Report and Form on Web Service**.
3. For Choose Service and Operation:
 - a. Web Service Reference - Select the Web service reference.
 - b. Operation - Select the method to be executed. For RESTful style Web references, doREST is selected automatically. The Operation option does not appear for Manual style Web references.
 - c. Click **Next**.
4. For Page and Region Attributes, review and update the page and region attributes and click **Next**.

 **Note:**

If the page you specify does not exist, the wizard creates the page for you.

5. For Input Items:
 - a. Identify which items to add to the form. To include an item, select **Yes** in the Create column. Otherwise, select **No**.
 - b. If necessary, edit the Item Name.
 - c. If necessary, edit the Item Label.
 - d. Click **Next**.
6. If applicable, specify the Item Names and Item Labels for basic authentication, then click **Next**.

 **Note:**

This step only appears if basic authentication was specified for this Web service reference when it was created.

7. Follow the on-screen instructions.
8. Click **Create**.

 **See Also:**

- ["Creating Web Service References Based on a WSDL"](#)
- ["Creating Web Service References Manually"](#)
- ["Creating RESTful Web Service References"](#)

Creating a Form on a Web Service

This section describes how to create a form on a Web service.

- [About the Create Form on Web Service Wizard](#)
- [Creating a Form by Adding a New Page](#)

About the Create Form on Web Service Wizard

The Create Form on Web Service Wizard creates a form and a submit button. You can execute this wizard after creating the Web service reference from a WSDL or on a RESTful style Web service, or by running the Create Page Wizard.

Use this wizard when you expect a scalar result from the Web service. A Web service that looks up a stock price is a good example because the input is a stock symbol and the output is the scalar value price.

Creating a Form by Adding a New Page

If you have an existing Web service reference, you can create a form based on a Web service reference by adding a new page.

To create a form by adding a new page:

1. Create the Web service reference.
2. Run the Create Page Wizard. On Create a Page:
 - a. For User Interface, if applicable select a user interface for the page.
This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.
 - b. For Page Type, select **Component** and then **Form**.
 - c. On Create page, select **Form on Web Service**.
3. For Page Attributes:
 - a. Page Number - Select a page to contain the new form, or enter a new page number.
 - b. Page Name - Specify a name for the page.
 - c. Page Mode - Identify the page mode.
To learn more, see field-level Help.
 - d. Page Group - Identify the name of the page group you would like to associate with this page. Page groups help developers manage the pages within an application. To create a page group, enter the name. To use an existing page group, select the name from the list of values.
 - e. Region Title - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - f. Breadcrumb - Enter a title for the region in which the form will appear.
 - g. Click **Next**.
4. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
5. For Choose Service:
 - a. Web Service Reference - Select the Web service reference.
 - b. Operation - Select the method to be executed. For RESTful style Web references, doREST is selected automatically. The Operation option does not appear for Manual style Web references.
 - c. Click **Next**.
6. For Input Items, make these selections and click **Next**:
 - a. Identify which items to add to the form. To include an item, select **Yes** in the Create column. Otherwise, select **No**.
 - b. If necessary, edit the Item Name.

- c. If necessary, edit the Item Label.
7. If applicable, specify the Item Names and Item Labels for basic authentication and click **Next**.

 **Note:**

This step only appears if basic authentication was specified for this Web service reference when it was created.

8. If applicable, specify the Item Names and Item Labels for basic authentication.
Note that this step only appears if basic authentication was specified for this Web service reference when it was created.
9. For Output Items, make these selections and click **Next**:
 - a. Identify which items to add to the form. To include an item, select **Yes** in the Create column. Otherwise, select **No**.
 - b. If necessary, edit the Item Name.
 - c. If necessary, edit the Item Label.
10. Click **Create**.

 **See Also:**

- ["Creating Web Service References Based on a WSDL"](#)
- ["Creating Web Service References Manually"](#)
- ["Managing Pages in a Database Application"](#)
- ["Managing the Application User Interface"](#)

Creating a Report on a Web Service

To create a report in which to display Web Service request results:

1. Create the Web service reference.
2. Run the Create Page Wizard. On Create a Page:
 - a. For User Interface, if applicable select a user interface for the page.
This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.
 - b. For Page Type, select **Component** and then **Report**.
 - c. On Create page, select **Report on Web Service**.
3. For Page Attributes:
 - a. Page Number - Select a page to contain the new form, or enter a new page number.
 - b. Page Name - Specify a name for the page.

- c. Page Mode - Identify the page mode.
To learn more, see field-level Help.
 - d. Breadcrumb - Enter a title for the region in which the form will appear.
 - e. Click **Next**.
4. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
5. If Web Reference Type is **REST**:
 - a. Web Reference Type - Choose **REST**.
 - b. Web Service Reference - Select the name of the Web service reference and click **Next**.
 - c. Result Stored in Collection - Enter the name of the temporary result set name where the Web service result is stored. The temporary result set is stored in an Application Express collection.
 - d. Report Template - Choose the report template to be used by this report region.
 - e. Rows Per Page- Select the number of rows to display per page.
 - f. Select the parameters to be included in the report.
 - g. Click **Next**.
6. For Web Reference Type, choose **REST**, **Generated from WSDL**, or **Manually Created**.

What appears next depends upon your selection.
7. If Web Reference Type is **Generated from WSDL**:
 - a. Web Reference Type - Choose **Generated from WSDL**.
 - b. Web Service Reference - Select the name of the Web service reference.
 - c. Operation - Select the operation. The operation is the method that will be executed..
 - d. Result Tree to Report On - Select the portion of the resulting XML document that contains the information you want to include in the report.
 - e. Report Parameters and Options:
 - Temporary Result Set Name - Enter a name for the collection that stores the Web service result.
 - Select and deselect the appropriate parameters.
 - f. Click **Next**.
8. If Web Reference Type is **Manually Created**:
 - a. Web Reference Type - Choose **Manually Created**.
 - b. Web Service Reference - Select the name of the Web service reference.
 - c. SOAP Style - Choose the style of the SOAP response. The style can be determined from the style attribute of the soap:binding element from the WSDL for this operation.

- d. Message Format - Choose the message format. The encoding can be determined from the use attribute of the soap:body element of the output message in the WSDL for this operation.
 - e. Result Node Path (XPath)(Value Required)- Enter the XPath expression to the node of the response message that you are creating the report on. The node is a child of the soap:Body element.

For example:

```
/result/myNode
```
 - f. Message Namespace - Enter the namespace for the response message. This can typically be determined by looking for the targetNamespace attribute of the WSDL and click **Next**.
 - g. Enter the name of the collection where the response message is stored.
 - h. Enter the names of the parameters that you want to be included in the report.
 - i. Click **Next**.
9. Click **Create**.

How to Invoke a Web Service as a Process

You can invoke a Web service as a process on the page. The process is created for you if you run one of the Create Form wizards for Web services. Running the process submits the request to the service provider. You can then display the request results in the report.

- [Invoking a Web Service as a Process](#)
- [Editing a Web Service Process](#)

Invoking a Web Service as a Process

To invoke a Web service as a process:

1. Create a Web Service Reference.
2. Run the Create Page Wizard. On Create a Page:
 - a. For User Interface, if applicable select a user interface for the page.

This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.
 - b. For Page Type, select **Component** and then **Blank Page**.
3. For Page Attributes:
 - a. Page Number - Select a page to contain the new form, or enter a new page number.
 - b. Page Name - Specify a name for the page.
 - c. Page Mode - Identify the page mode.

To learn more, see field-level Help.
 - d. Page Group - Identify the name of the page group you would like to associate with this page. Page groups help developers manage the pages within an application. To create a page group, enter the name. To use an existing page group, select the name from the list of values.

- e. Breadcrumb - Enter a title for the region in which the form will appear.
 - f. Click **Next**.
4. For Navigation Preference:
- a. Select **Do not associate this page with a navigation menu entry**.
 - b. Click **Next**.

5. Click **Finish**.

Page Designer appears.

6. Under Page Rendering or Page Processing, locate the Processes section, right-click and select **Create Process**.

The New process appears.

7. Edit the process attributes in the Property Editor.

Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.



Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

8. Edit **Identification** attributes:

- a. Find the **Identification** group.
- b. Name - Enter the name of the page process for easy identification by developers.
- c. Type - Select **Web Service**.

9. Edit **Settings**:

- a. Find the **Settings** group.
- b. Web Service Operation - Select a Web Service Operation.
- c. Store Result In - Select **Item(s)** or **Collection**.

10. Edit **Execution Options** Options:

- a. Find the **Execution Options** group.
- b. Sequence - Specify the sequence for this component. The sequence determines the order of execution.
- c. Point - Select the processing point.

11. Edit **Success Message**:

- a. Find the **Success Message** group.
- b. Success Message - Enter the success message for this process. If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed.

To learn more, click the **Help** tab.

12. Edit **Error Message**:

- a. Find the **Error Message** group.
- b. Error Message - Enter the error message for this process. This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays..

To learn more, click the **Help** tab.

13. Edit Server-side Condition:

- a. Find the **Server-side Condition** group.
- b. When Button Pressed - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.
- c. Condition Type - Select a condition type that must be met in order for this component to be processed. Depending upon your selection, additional attributes may appear.

14. Click Save.



See Also:

- ["Creating Web Service References Based on a WSDL"](#)
- ["Creating Web Service References Manually"](#)
- ["Creating a RESTful Web Service Reference"](#)
- ["Managing Pages in a Database Application"](#)

Editing a Web Service Process

After you create a process of type Web service on a Web service reference created from a WSDL or a RESTful style Web reference, you can change the attributes of the input and output parameters to the service.

To edit a Web service process:

1. Create a Web service process.
2. Navigate to the page containing the Web service process.
Page Designer appears.
3. In the left pane, find the process name and expand the tree.
In Parameters and **Out Parameters** display under the process.
4. Expand **In Parameters** and **Out Parameters**. Defined parameters appear.
5. Select a parameter and edit the attributes in the Property Editor.

Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.



Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. Click **Save**.



See Also:

["How to Invoke a Web Service as a Process"](#)

Viewing a Web Service Reference History

The Web Services History displays changes to Web service references for the current application by application ID, Web service references name, developer, and date.

To view a history of Web service reference changes:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Data Sources, click **Legacy Web Service References**.
The Web Service References page appears.
3. Click **History**.



Note:

The History button only appears on the Web Service Reference page if at least one reference exists.

Accessing Data with Database Links

Access data remotely by creating a database link.

- [About Database Links](#)
- [Creating a Database Link](#)
- [Viewing an Existing Database Link](#)

 **See Also:**

- "Managing Database Objects with Object Browser" in *Oracle Application Express SQL Workshop Guide*
- "Database Links" in *Oracle Database Administrator's Guide*

About Database Links

Because Oracle Application Express runs in the Oracle database, you have access to all distributed Oracle database capabilities. Typically, you perform distributed database operations using database links.

A database link is a schema object in one database that enables you to access objects on another database. Once you have created the database link you can access the remote objects by appending `@dblink` to the table or view name where `dblink` is the Database Link Name you specify in the Create Database Object Wizard.

 **Note:**

By default, the CREATE DATABASE LINK system privilege is not granted to a provisioned workspace or database user. To use this feature, a DBA or administrator must grant this specific privilege to the database user in the user's workspace. See "Creating Database Links" in *Oracle Database Administrator's Guide*.

Creating a Database Link

To create a database link:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**. Object Browser home page appears.
2. Click **Create**.
3. Select **Database Link** and click **Next**.
4. Follow the on-screen instructions.

Note that Database Link names must conform to Oracle naming conventions and cannot contain spaces, or start with a number or underscore.

Viewing an Existing Database Link

To view an existing a database link:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**. Object Browser appears.
2. Select the object type **Database Links** at the top of the page.

About DML Locking

Use automatic Data Manipulation Language (DML) in Oracle Application Express to update or delete rows of a table.

- [About DML Locking](#)
- [APEX_DML_LOCK_WAIT_TIME](#)
- [FSP_DML_LOCK_ROW](#)

About DML Locking

When you use automatic Data Manipulation Language (DML) in Oracle Application Express to update or delete rows of a table, a transaction is initiated to first lock the row, verify if it has changed since it was displayed on the page, and then finally issue the actual `UPDATE` or `DELETE` statement for the row.

In some environments where locking of rows is prevalent, you may want to control the DML operation and determine if the DML operation:

- waits indefinitely
- fails immediately
- waits for a specified period of time

APEX_DML_LOCK_WAIT_TIME

You can set the value of an application substitution string, an application item, or a page item to `APEX_DML_LOCK_WAIT_TIME` to control the DML operation.

`APEX_DML_LOCK_WAIT_TIME` supports the following values:

- `NULL` (the default), results in the same behavior as previous versions of Oracle Application Express, that is, wait indefinitely.
- `0` fails immediately if the row is locked by another database session.
- `> 0` and the row is locked, waits for the specified number of seconds.

When set in an application, the value for `APEX_DML_LOCK_WAIT_TIME` applies to all `UPDATE` and `DELETE` DML operations using Automatic DML in the entire application. To control a specific Automatic DML process, update the value of `APEX_DML_LOCK_WAIT_TIME` before the Automatic DML process and reset it after the Automatic DML process. Note that this does not affect updates and deletes using tabular forms.

FSP_DML_LOCK_ROW

You can also set the value of an application substitution string, an application item, or a page item to `FSP_DML_LOCK_ROW` to control the DML operation. `FSP_DML_LOCK_ROW` supports the following values:

- If the value is set to `FALSE`, then no `SELECT FOR UPDATE` is issued.

- If the value is anything other than `FALSE`, the default behavior of `SELECT FOR UPDATE` is performed when issuing an `UPDATE` or `DELETE` DML operation using Automatic DML.

19

Extending Application Capabilities

Learn about how to extend application capabilities including managing email, running background PL/SQL, utilizing plug-ins, and understanding jQuery support.

- [Sending Email from an Application](#)
To configure and manage email in Oracle Application Express, an Instance administrator must log in to Oracle Application Express Administration Services.
- [Implementing Plug-ins](#)
Create plug-ins to declaratively extend, share, and reuse the built-in types available with Oracle Application Express.
- [Understanding jQuery and jQuery UI Support](#)
Oracle Application Express includes the jQuery 3.1.1 and jQuery UI 1.12.x.
- [Manually Refreshing Oracle Application Express Components](#)
Trigger the `apexrefresh` event for the relevant component.

See Also:

- [Oracle Application Express API Reference](#)
- ["Deploying an Application"](#)

Sending Email from an Application

To configure and manage email in Oracle Application Express, an Instance administrator must log in to Oracle Application Express Administration Services.

- [About Configuring Email](#)
To enable Oracle Application Express to send email, an Instance administrator must configure email settings.
- [About Email Provisioning](#)
You can configure Oracle Application Express to automatically email users their login credentials when a new workspace request has been approved. To accomplish this, you must complete the email configuration process and select the provisioning status, Request with Email Verification.
- [About the Mail Queue](#)
Instance administrators can manage email sent from applications by monitoring email messages in the mail queue and mail log.
- [About the APEX_MAIL Package](#)
You can send email from an Oracle Application Express application using the `APEX_MAIL` package.

About Configuring Email

To enable Oracle Application Express to send email, an Instance administrator must configure email settings.

- [About Enabling Network Services](#)
- [About Configuring Email in a Full Development Environment](#)



See Also:

"Configuring Email" in *Oracle Application Express Administration Guide*

About Enabling Network Services

By default, the ability to interact with network services is disabled in Oracle Database 11g Release 1 or 2 or later. Therefore, if you are running Oracle Application Express with Oracle Database 11g Release 1 or 2 or later, you must use the new `DBMS_NETWORK_ACL_ADMIN` package to grant connect privileges to any host for the `APEX_180200` database user. Failing to grant these privileges results in issues with:

- Sending outbound mail in Oracle Application Express.
- Using Web services in Oracle Application Express.
- PDF/report printing.



See Also:

- "Enabling Network Services in Oracle Database 11g or Later" for your installation scenario in *Oracle Application Express Installation Guide*:
 - "Installing Application Express and Configuring Oracle REST Data Services"
 - "Installing Application Express and Configuring Embedded PL/SQL Gateway"
 - "Installing Application Express and Configuring Oracle HTTP Server"
- "Configuring Email" in *Oracle Application Express Administration Guide*

About Configuring Email in a Full Development Environment

To configure Oracle Application Express to send mail in a full development environment your Instance administrator must log in to Oracle Application Express Administration Services, navigate to the Instance Settings page and configure Email attributes.

 **See Also:**

"Configuring Email" in *Oracle Application Express Administration Guide*

About Email Provisioning

You can configure Oracle Application Express to automatically email users their login credentials when a new workspace request has been approved. To accomplish this, you must complete the email configuration process and select the provisioning status, Request with Email Verification.

 **See Also:**

- "Configuring Email" in *Oracle Application Express Administration Guide*
- "About Specifying How Workspaces Are Created" in *Oracle Application Express Administration Guide*
- "Selecting a Provisioning Mode" in *Oracle Application Express Administration Guide*

About the Mail Queue

Instance administrators can manage email sent from applications by monitoring email messages in the mail queue and mail log.

 **See Also:**

"Managing the Mail Queue" in *Oracle Application Express Administration Guide*

About the APEX_MAIL Package

You can send email from an Oracle Application Express application using the `APEX_MAIL` package.

The `APEX_MAIL` package is built on top of the Oracle supplied `UTL_SMTP` package. Because of this dependence, the `UTL_SMTP` package must be installed and functioning to use `APEX_MAIL`.

 **See Also:**

- "APEX_MAIL" in *Oracle Application Express API Reference*
- "Configuring Email" in *Oracle Application Express Administration Guide*

Implementing Plug-ins

Create plug-ins to declaratively extend, share, and reuse the built-in types available with Oracle Application Express.

- [About Plug-ins](#)
- [Accessing the Plug-ins Page](#)
- [Creating a Plug-in](#)
- [Editing a Plug-in](#)
- [Adding Custom Attributes to a Plug-in](#)
- [Uploading Files Associated with a Plug-in](#)
- [Adding Events to a Plug-in](#)
- [Deleting a Plug-in](#)
- [Viewing the Plug-in Repository](#)
- [Importing a Plug-in from the Plug-in Page](#)
- [Exporting a Plug-in from the Plug-in Page](#)
- [Resetting the Plug-in Interactive Report](#)
- [Viewing Plug-in Utilization Page](#)
- [Viewing Plug-in History](#)

About Plug-ins

Oracle Application Express supports a set group of authentication scheme, authorization scheme, item, region, dynamic action, and process types. Plug-ins offer a means of augmenting these built-in types by declaratively creating and using new types in your application. Because plug-ins are designed for reuse, developers can export and import them to other workspaces and also share them with the Oracle Application Express Plug-in community by using the Plug-in Repository.

The process of implementing a plug-in involves the following steps:

1. Create a plug-in or import a plug-in into your application workspace.
2. Edit or create an authorization scheme, item, region, process, or dynamic action type to use the plug-in.
3. Run your application to test the plug-in.

Viewing Plug-in Examples

You can view plug-in implementation examples in the following locations:

- Go to the Oracle Plug-in Repository.
See "[Viewing the Plug-in Repository](#)."
- Install the sample application, *Sample Database Application*.
See "[Managing Productivity and Sample Apps in a Full Development Environment](#)" and "[Installing and Running Sample Database Application](#)."
- Go to the Oracle Learning Library at <http://www.oracle.com/oll/apex>. Enter search criteria in the field provided and click **Search**.

Accessing the Plug-ins Page

To access the Plug-ins page

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Other Components, click **Plug-ins**.

The Plug-ins page displays with the Plug-ins tab selected by default. All available plug-ins appear.

You can customize the appearance of the page using the Search bar at the top of the page.

Creating a Plug-in

To create a plug-in:

Tip:

To learn more about an attributes described in this section see field-level Help. See "[Viewing Field-level Help in Oracle Application Express](#)."

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Other Components, click **Plug-ins**.
3. Click **Create**.

The Create Plug-in wizard appears.

4. For Create Plug-in, select the method by which you would like to create a plug-in and click **Next**.

5. Under Name:
 - a. Name (Required) - Enter name of the plug-in.
 - b. Internal Name (Required) - Enter the internal name of the plug-in. This name must be unique within the current application.

 **Note:**

To insure the internal name is a globally unique name worldwide, Oracle recommends that your organization domain name be used as a prefix to internal plug-in names. For example, a domain name of `example.com.com` prefixed to a plug-in named `Slider`, would result in an internal name of `COM.EXAMPLE.SLIDER`.

- c. Type (Required) - Select the type of component that can use this plug-in. Depending upon the plug-in type you select, the options under Callbacks and Standard Attributes differ. To learn more, see field-level Help.
 - d. Category - Only displays if the selected type is Dynamic Action. Select the category the plug-in is displayed under on the user interface.
6. Under Subscription:
 - Reference Master Plug-in From - To base this plug-in on another plug-in in this workspace, select the plug-in from the list. Otherwise, leave the field blank to make this the master copy of this plug-in.
7. Under Source:
 - a. PL/SQL Code - Enter a PL/SQL anonymous block of code that contains the procedures for rendering, validating, executing, and performing Ajax callbacks for this plug-in. For performance reasons you can also store this code in a PL/SQL package in the database.
 - b. Do not validate PL/SQL code (parse PL/SQL code at runtime only) - Select this option to parse the PL/SQL code at runtime only. Otherwise, the code is parsed when the plug-in is created.
8. Under Callbacks, configure that appropriate attributes. The attributes that display depend upon the plug-in type. To learn more about an attribute and view examples, see field-level Help.

 **Tip:**

All Callback function names can reference a function of the anonymous PL/SQL code block, a function within a package or a standalone function in the database.

9. Under User Interfaces, select the display devices the App Builder must support for this plug-in. Options include:
 - **Desktop**
 - **Mobile**
10. For Standard Attributes, select the attributes that apply to this plug-in. Standard Attributes do not display for some plug-ins. To learn more, see field-level Help.

11. Under Information:
 - a. Version - Enter a string to identify the plug-in version.
 - b. About URL - Enter a URL to the plug-in authors home page or to additional information about the plug-in.
12. For Help Text, enter help text used by the user to understand how the plug-in works.
13. For Comments, enter comments and notes that never display when the application is running.

To learn more about each option, see field-level Help.
14. Click **Create. Plug-in**.

Now that the plug-in is created, you can specify additional custom attributes, upload files such as image, CSS and JavaScript files to associate with your plug-in and add events.

 **See Also:**

- ["Adding Custom Attributes to a Plug-in"](#)
- ["Uploading Files Associated with a Plug-in"](#)
- ["Adding Events to a Plug-in"](#)

Editing a Plug-in

To edit a plug-in:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.

The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to edit or view.

The Plug-in Create/Edit page appears.
4. Make modifications.

To learn more about each option, see field-level Help.
5. Click **Apply Changes**.

Adding Custom Attributes to a Plug-in

Custom Attributes specified by the developer might contain items referenced with substitution syntax.

To add custom attributes to the plug-in:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to modify.
The Plug-in Create/Edit page appears.
4. Under Custom Attributes, enable or disable the substitution of attribute values.
 - **Substitute Attribute Values** - Custom attribute values specified by the developer might contain items referenced with substitution syntax, for example `&P1_DNAME`.

If set to **Yes**, Application Express automatically replaces substitution syntax with their actual values.

If set to **No**, substitution syntax is written unchanged into the `attribute_01` through `attribute_15` record type attributes of `p_plugin`, `p_item`, `p_region`, and so on. The plug-in developer is responsible for replacing those substitution syntax references with a call to `apex_plugin_util.replace_substitutions` or perform similar replacements. See item help for further details.

To learn more, see field-level Help.
5. To add an attribute click **Add Attribute**.
The Edit Attribute page appears. Edit the appropriate attributes.
To learn more about a specific attribute, see field-level Help.
6. Click **Create** to create the attribute and go back to the Edit page, or click **Create and Create Another** to create the attribute and continue to create another attribute.

**Note:**

If you click **Create** or **Create and Create Another** and the **Return To Page** check box on the right panel under Plug-ins is checked, this same Edit Attribute page displays.

Uploading Files Associated with a Plug-in

Learns how to upload the files associated with a plug-in.

- [Uploading a File to Associate with a Plug-in](#)
- [Automatically Loading CSS and JavaScript Files](#)

Uploading a File to Associate with a Plug-in

To upload a file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Select the plug-in.
The Plug-in Create/Edit page appears.
4. Locate the **Files** section.
File Prefix determines the virtual path the Web server uses to point to the files of the plug-in. Do not specify anything to reference files which are stored with your plug-in definition in the database.
5. To upload a file:
 - a. Click **Upload File**.
 - b. Browse to and select the file you want to upload.
 - c. Click **Upload**.
The Create/Edit page appears. The name of the uploaded file appears under Files.
6. Click **Apply Changes**.

Automatically Loading CSS and JavaScript Files

You can have Oracle Application Express automatically load CSS and JavaScript files when a plug-in is used on a page by configuring the File URLs to Load attributes. To specify which of the uploaded files should be loaded and in what order.

To automatically load a CSS or JavaScript file:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Select the plug-in.
The Plug-in Create/Edit page appears.
4. Under File URLs to Load:
 - a. Cascading Style Sheet - Enter Cascading Style Sheet file URLs to be loaded with this plug-in.
 - b. JavaScript - Enter JavaScript file URLs for code to be loaded with this plug-in.

You can substitute with the value of the plug-in's file prefix by using substitution string `#PLUGIN_FILES#`. For example:

```
#PLUGIN_FILES#my_plugin.css
```

To learn more and view examples, see field-level Help.

5. Click **Apply Changes**.

Adding Events to a Plug-in

This section describes how to add events to an item, region, or dynamic action type plug-in, enables them to be exposed to dynamic actions. For example, a Slider plug-in that exposes events such as Start Slide, Sliding, and Stop Slide, allows the creation of dynamic actions that can react when these events occur.

To add events to a plug-in:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to edit.
The Plug-in Create/Edit page appears
4. Under Events and click **Add Event**.
A new row displays under Events.
5. Under Events:
 - a. Name - The display name under which the plug-in event appears in the dynamic action, for example: `Start Slide`.
 - b. Internal Name - The name of the assigned JavaScript event that triggers the dynamic action, for example: `slidestart`.
6. Click **Add Event**.
7. Repeat steps 3 through 4 to add another event.
8. Click **Apply Changes**.



See Also:

["Accessing the Plug-ins Page"](#)

Deleting a Plug-in

You can delete a plug-in if it is not in use. If it is in use, the Delete button does not display.

To delete a plug-in:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.

Resetting the Plug-in Interactive Report

You can reset the plug-in interactive report to clear all current filters applied to the report.

To reset the interactive report:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. On the Plug-ins page, click **Reset**.

Viewing Plug-in Utilization Page

The Plug-in Utilization page displays which pages, components, and regions use each plug-in.

To view plug-in utilization:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click **Utilization**.
The Utilization page appears.

Viewing Plug-in History

The Plug-in History page shows the actions taken on each plug-in, the developer that performed the action and the date of each action.

To view plug-in history:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click **History**.

The History page appears.

Understanding jQuery and jQuery UI Support

Oracle Application Express includes the jQuery 3.1.1 and jQuery UI 1.12.x.

- [About Upgrading to jQuery 3.1.1](#)
- [About Upgrading to jQuery UI 1.12.x](#)

About Upgrading to jQuery 3.1.1

jQuery 3.1.1 breaks compatibility with earlier 2.x versions. If your application relies on removed jQuery 2.x functionality, you can use the jQuery Migrate plug-in. To include the jQuery Migrate plug-in, go to Desktop User Interface Details and set **Include jQuery Migrate** to **Yes**.

If your application relies on removed 1.x jQuery APIs, be aware that this functionality no longer works as of Oracle Application Express release 18.1. You must update the JavaScript to only use jQuery 3.1 APIs. See the jQuery migration guides to learn more:

- To migrate from 1.x to 2.x see <http://jquery.com/upgrade-guide/1.9/>
- To migrate from 2.x to 3.x see <https://jquery.com/upgrade-guide/3.0/>



See Also:

["Editing User Interface Details"](#)

About Upgrading to jQuery UI 1.12.x

jQuery UI 1.12.x has changed the folder structure and files that make up the library compared to jQuery UI 1.10.4. If you have direct references to any of the old file filenames, you must update them to the new name. For example, if you previously referenced the tabs widget with `#JQUERYUI_DIRECTORY#ui/#MIN_DIRECTORY#jquery.ui.tabs#MIN#.js`, you must change it to `#JQUERYUI_DIRECTORY#ui/widgets/#MIN_DIRECTORY#tabs#MIN#.js`.

The Oracle Application Express specific bundle `jquery-ui-apex[.min].js` that loads by default for Application Express desktop UI pages (either as `jquery-ui-apex.js` or as part of `desktop[_all].min.js`) includes all the core files, the drop effect, and the following widgets:

- button
- checkboxradio
- controlgroup
- datepicker
- dialog
- draggable
- droppable

- resizable
- selectable
- sortable
- tooltip

This is essentially the same set as in 1.10.4 with the addition of sortable. If you have a separate reference to sortable, you can remove it.

The `jquery-ui-apex[.min].css` file loads by default and includes all the jQuery UI CSS files. If you had references to individual jQuery UI css files, you can remove them.

Manually Refreshing Oracle Application Express Components

Trigger the `apexrefresh` event for the relevant component.

Although the Dynamic Action framework provides refresh capability for supported Oracle Application Express components, there may be cases where you wish to manually invoke a refresh from JavaScript code. To do this, you can trigger the `apexrefresh` event for the relevant component, as follows:

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

Managing Application Security

Administrators are primarily responsible for ensuring the security of the Oracle Application Express installation and developers are responsible for building secure applications. Learn about how to implement best practices for Oracle Application Express application security.

- [Understanding Administrator Security Best Practices](#)
Learn about security best practices for Oracle Application Express administrators.
- [Understanding Developer Security Best Practices](#)
- [Controlling Access to Applications, Pages, and Page Components](#)
Control access to an application, individual pages, or page components by creating an access control list.
- [Establishing User Identity Through Authentication](#)
Use authentication to establish a user's identity to control access to an application. Authentication may require a user identify a user name and password or could involve the use of digital certificates or a secure key.
- [Providing Security Through Authorization](#)
Extend the security of your application by creating an authorization scheme.



See Also:

["Running Advisor to Check Application Integrity"](#)

Understanding Administrator Security Best Practices

Learn about security best practices for Oracle Application Express administrators.



Tip:

This section references many settings in the Oracle Application Express Administration Services application. Functionality in the Administration Services application is not available in Oracle Database Cloud Service (Database Schema).

- [About Oracle Application Express Administrator Roles](#)
Oracle Application Express includes two different types of administrators: Workspace administrators and Instance administrators.

- [About Configuring Oracle REST Data Services with Oracle Application Express](#)
Oracle recommends using Oracle REST Data Services with Oracle Application Express.
- [About Configuring Oracle HTTP Server with mod_plsql with Oracle Application Express](#)
Oracle HTTP Server uses the `mod_plsql` plug-in to communicate with the Oracle Application Express engine within the Oracle database. `mod_plsql` functions act as a communication broker between the web server and the Oracle Application Express engine in the Oracle database.
- [About Security Considerations When Using the Embedded PL/SQL Gateway](#)
The embedded PL/SQL gateway runs in the database as part of the Oracle XML DB HTTP listener. The Oracle XML DB HTTP listener and embedded PL/SQL gateway provides the equivalent core features of Oracle HTTP Server and `mod_plsql`.
- [Managing Instance Security](#)
Learn about security best practices when managing an Oracle Application Express instance.
- [Configuring Instance Settings](#)
Security best practices when configuring an Oracle Application Express instance include determining whether to automate the creation workspaces., enabling and configuring email, configuring storage options, and creating wallets.
- [About Configuring Workspace Purge Settings](#)
Save storage space and improve system performance by purging inactive workspaces.
- [Understanding Workspace Management](#)
Instance administrators can define associations between workspaces and schemas. Administrators can manage developers and users at the workspace or instance-level and control user access to key components such App Builder, Team Development, and SQL Workshop, and PL/SQL editing.
- [About Integrating with Oracle BI Publisher](#)
Oracle recommends that you employ Secure Sockets Layer (SSL) when integrating Oracle Application Express with Oracle BI Publisher. Once SSL (HTTPS protocol) is configured for your Oracle BI Publisher server, you must create a wallet and specify the HTTPS protocol for the Report Server in the internal administration Environment Settings.
- [About the Advantages of the Application Express Runtime Environment](#)
An Oracle Application Express runtime environment enables you to run production applications, but does not provide a web interface for administration or direct development of these applications
- [Enabling Network Services in Oracle Database 11g or Later](#)
You must enable network services in Oracle Database 11g or later versions to send outbound mail, use Web services, or use PDF report printing in Oracle Application Express.



See Also:

["Understanding Developer Security Best Practices"](#)

About Oracle Application Express Administrator Roles

Oracle Application Express includes two different types of administrators: Workspace administrators and Instance administrators.

To access the Oracle Application Express development environment, users log in to a shared work area called a workspace. **Workspace administrators** are users who perform administrator tasks specific to a workspace such as managing user accounts, monitoring workspace activity, and viewing log files. **Instance administrators** are superusers that manage an entire hosted instance using the Application Express Administration Services application.

Tip:

The role of Instance administrator is not available in Oracle Database Cloud Service (Database Schema).

See Also:

"Workspace and Application Administration" and "Oracle Application Express Administration Services" in *Oracle Application Express Administration Guide*

About Configuring Oracle REST Data Services with Oracle Application Express

Oracle recommends using Oracle REST Data Services with Oracle Application Express.

Oracle REST Data Services (formerly known as Oracle Oracle Application Express Listener) is a J2EE application which communicates with the Oracle Database by mapping browser requests to the Application Express engine database over a SQL*Net connection. Oracle REST Data Services is the strategic direction for Oracle Application Express and Oracle recommends using it in practically all circumstances. In a production environment, you deploy Oracle REST Data Services web archive files to a supported Java EE application server, like Oracle Web Logic Server. Each deployment can be configured individually and serves the same purpose as a `mod_plsql` Database Access Descriptor, which is to communicate with an Oracle database.

An Oracle REST Data Services deployment configuration contains several security related parameters. In a configuration for Oracle Application Express, Oracle recommends to set the parameter `security.requestValidationFunction` to `wwv_flow_epg_include_modules.authorize`. This activates the white list of callable procedures which ships with Oracle Application Express and prohibits calls to other procedures. This can be extended using the validation functions shipped with Oracle Application Express.

 **See Also:**

"Restricting Access to Oracle Application Express by Database Access Descriptor (DAD)" in *Oracle Application Express Administration Guide*

About Configuring Oracle HTTP Server with `mod_plsql` with Oracle Application Express

Oracle HTTP Server uses the `mod_plsql` plug-in to communicate with the Oracle Application Express engine within the Oracle database. `mod_plsql` functions act as a communication broker between the web server and the Oracle Application Express engine in the Oracle database.

 **Tip:**

`mod_plsql` is deprecated as of Oracle HTTP Server 12c (12.1.3). For more information about this deprecation, please see My Oracle Support Note 1576588.1. Oracle recommends using Oracle REST Data Services instead.

Each `mod_plsql` request is associated with a set of configuration values used to access the database called a Database Access Descriptor (DAD). `mod_plsql` provides a DAD parameter called `PlsqlRequestValidationFunction` which enables you to allow or disallow further processing of a requested procedure. You can utilize this parameter to implement tighter security for your PL/SQL application by blocking package and procedure calls which should not be allowed to run from the DAD. Oracle recommends a DAD configuration for Oracle Application Express which utilizes the `PlsqlRequestValidationFunction` directive with a value of `www_flow_epg_include_modules.authorize`.

The purpose of the `PlsqlRequestValidationFunction` parameter is to control which procedures can be invoked through `mod_plsql`. By default, the only procedures permitted are the public entry points of Oracle Application Express. This can be extended using the validation functions shipped with Oracle Application Express.

 **See Also:**

"Restricting Access to Oracle Application Express by Database Access Descriptor (DAD)" in *Oracle Application Express Administration Guide*

About Security Considerations When Using the Embedded PL/SQL Gateway

The embedded PL/SQL gateway runs in the database as part of the Oracle XML DB HTTP listener. The Oracle XML DB HTTP listener and embedded PL/SQL gateway provides the equivalent core features of Oracle HTTP Server and `mod_plsql`.

Because the HTTP Listener runs in the same database where Oracle Application Express is installed, it is not possible to separate the HTTP listener from the database. For this reason, Oracle does not recommend the embedded PL/SQL gateway for applications that run on the Internet or for production applications. Oracle recommends using Oracle REST Data Services instead. Additionally, the embedded PL/SQL gateway does not provide the same flexibility of configuration and detailed logging as Oracle REST Data Services.

Managing Instance Security

Learn about security best practices when managing an Oracle Application Express instance.

- [About Creating Login Controls](#)
Administrators can configure login controls for an entire instance or for individual workspaces.
- [About Enabling Public File Upload](#)
Administrators can use the Allow Public File Upload attribute to control whether unauthenticated users can upload files in applications that provide file upload capability
- [About Restricting User Access by IP Address](#)
Administrators can restrict user access to an Oracle Application Express instance by entering a comma-delimited list of allowable IP addresses in the Restrict Access by IP Address attribute on the Manage Instance, Security page.
- [About Specifying an Instance Proxy](#)
Administrators can use the Instance Proxy attribute to configure an entire Oracle Application Express instance to use a proxy for all outbound HTTP traffic.
- [About Utilizing Secure Sockets Layer](#)
Secure Sockets Layer (SSL) is a protocol for managing the security of data transmitted on the Internet. For web applications, SSL is implemented by using the HTTPS protocol. Oracle recommends that you run Oracle Application Express applications using SSL (HTTPS protocol) to prevent any sensitive data from being sent over an unencrypted (cleartext) communication channel.
- [About Enabling RESTful Access](#)
Administrators can use the Allow RESTful Access attribute to control whether developers can expose report regions as RESTful services.
- [About Rejoin Sessions](#)
Use the Rejoin Sessions attribute to control if Oracle Application Express supports application URLs that do not contain session IDs.
- [About Isolating Workspaces](#)
Isolating workspaces is an effective approach to preventing browser attacks.

- [About Utilizing Session Timeout](#)
By configuring Session Timeout attributes, you can reduce your application's exposure.
- [Restricting Password Reuse](#)
Site administrators can restrict password reuse for all administrator, developer, and end-user accounts based on a history of passwords previously used for the accounts.
- [About Using SQL in Websheets](#)
Disabling the Allow SQL Worksheet attribute enables administrators to prevent Worksheet users from accessing underlying database objects using SQL tags or from creating SQL reports.
- [About Enabling RESTful Services](#)
Administrators can control the ability to create and access RESTful Services at either the workspace or instance-level.



See Also:

"Configuring Security Settings" in *Oracle Application Express Administration Guide*

About Creating Login Controls

Administrators can configure login controls for an entire instance or for individual workspaces.

Workspace administrators and developers can create user accounts for the purpose of logging in to the Oracle Application Express development environment and for end-user authentication to applications developed within their workspaces.

Administrators can configure login controls at the instance or workspace level. For example, if an Instance administrator configures account login controls in Oracle Application Express Administration Services that configuration applies to all Application Express accounts in all workspaces across an entire development instance.

If the Instance administrator does *not* enable login controls across an entire instance, then each Workspace administrator can enable the following controls on a workspace-by-workspace basis:

- End-user account expiration and locking.
- A maximum number of failed login attempts for end-user accounts.
- The password lifetime (or number of days an end-user account password can be used before it expires for end-user accounts).

 **See Also:**

- “Configuring Authentication Controls for an Instance” in *Oracle Application Express Administration Guide*
- “Creating Account Login Controls for a Workspace” in *Oracle Application Express Administration Guide*

About Enabling Public File Upload

Administrators can use the Allow Public File Upload attribute to control whether unauthenticated users can upload files in applications that provide file upload capability

 **See Also:**

“Controlling Public File Upload” in *Oracle Application Express Administration Guide*

About Restricting User Access by IP Address

Administrators can restrict user access to an Oracle Application Express instance by entering a comma-delimited list of allowable IP addresses in the Restrict Access by IP Address attribute on the Manage Instance, Security page.

 **See Also:**

“Restricting User Access by IP Address” in *Oracle Application Express Administration Guide*

About Specifying an Instance Proxy

Administrators can use the Instance Proxy attribute to configure an entire Oracle Application Express instance to use a proxy for all outbound HTTP traffic.

Setting a proxy at the instance-level supersedes any proxies defined at the application-level or in web service references. If a proxy is specified, regions of type URL, Web services, and report printing will use the proxy.



See Also:

"Configuring a Proxy Server for an Instance" in *Oracle Application Express Administration Guide*

About Utilizing Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol for managing the security of data transmitted on the Internet. For web applications, SSL is implemented by using the HTTPS protocol. Oracle recommends that you run Oracle Application Express applications using SSL (HTTPS protocol) to prevent any sensitive data from being sent over an unencrypted (cleartext) communication channel.

Instance administrators can configure both their Oracle Application Express instance and all related applications to require HTTPS by configuring the **Require HTTPS** and **Require Outbound HTTPS** attributes. Configuring these attributes forces authenticated pages within the App Builder to require HTTPS which encrypts network communications. Changing the Require HTTPS option does not affect the web server in any way. You will still need to enable the web server for HTTPS for these options to have any meaning. To learn more about enabling HTTPS, see the documentation for your specific web server.

Require HTTPS options include:

- **Always** - Enforces HTTPS for all applications (including the Oracle Application Express development and administration applications) to require HTTPS.
- **Development and Administration** - Forces all internal applications within Oracle Application Express (that is, App Builder, SQL Workshop, Instance Administration and so on) to require HTTPS.
- **Application specific** - Makes HTTPS dependent on application-level settings.



See Also:

"Configuring HTTP Protocol Attributes" in *Oracle Application Express Administration Guide*

About Enabling RESTful Access

Administrators can use the Allow RESTful Access attribute to control whether developers can expose report regions as RESTful services.



See Also:

"Enabling REST for an Instance" in *Oracle Application Express Administration Guide*

About Rejoin Sessions

Use the Rejoin Sessions attribute to control if Oracle Application Express supports application URLs that do not contain session IDs.

Tip:

For security reasons, Oracle recommends that administrators disable Rejoin Sessions unless they implement workspace isolation by configuring the Allow Hostname attribute at the workspace or instance-level. See "[About Isolating Workspaces](#)."

When rejoin sessions is enabled, Oracle Application Express attempts to use the session cookie to join an existing session, when a URL does not contain a session ID. To use Rejoin Sessions, administrators must enable Rejoin Sessions at the application or page-level. Note that a more restrictive value of Rejoin Sessions at the instance-level overrides application and page settings.

While supporting session joins is convenient, it does present some serious security risks:

- **Triggering malicious session state changes or other modifications.**

If an attacker tricks the user into clicking an application link, this can trigger malicious session state changes or other modifications. Because of this risk, Application Express requires a checksum that the attacker will not be able to guess when processing requests that alters data.

- **Triggering unintended changes between applications.**

If a user has two applications open in separate tabs on the same Oracle Application Express instance, one application could trigger unintended changes in the other one. The attacking application could be written by a hacker or it could have an XSS vulnerability that enables an attacker to inject code. This application could use Ajax calls that simulate user interactions with the other application.

See Also:

- "[About Enabling Support for Bookmarks](#)"
- "[Browser Security](#)"
- "[Configuring Rejoin Sessions](#)"
- "Configuring Rejoin Sessions for an Instance" in *Oracle Application Express Administration Guide*

About Isolating Workspaces

Isolating workspaces is an effective approach to preventing browser attacks.

The only way to truly isolate a workspace is to enforce different domains in the URL by configuring the Allow Hostnames attribute in Oracle Application Express Administration Services. When the URLs of the attacker and the victim have different domains and hostnames, the browser's same-origin policy prevents attacks.

Workspace isolation by configuring Allow Hostnames is a counter measure against client side attacks that attempt to cross workspace boundaries. This security measure is not necessary if you trust all applications that are accessible using the instance's host which includes applications that are written in other frameworks and languages such as Oracle Application Development Framework (ADF) and Java.

 **See Also:**

- "Isolating a Workspace to Prevent Browser Attacks" in *Oracle Application Express Administration Guide*
- "Isolating All Workspaces in an Instance" in *Oracle Application Express Administration Guide*

About Utilizing Session Timeout

By configuring Session Timeout attributes, you can reduce your application's exposure.

Users often leave their computers unattended for extended periods and do not close applications before departing. Therefore, an unauthorized person can easily assume the user's identity within the application. By setting the session and idle timeout, users are automatically logged out of their application after the specified timeout.

Session Timeout attributes include:

- Maximum Session Length in Seconds
- Session Timeout URL
- Maximum Session Idle Time in Seconds
- Idle Timeout URL

Developers can configure session time out for a specific application, or administrators can configure sessions time out for an entire instance.

 **See Also:**

- "[Session Management](#)" to learn how to configure these attributes at an application-level.
- To configure these attributes for an entire Oracle Application Express instance, see "Configuring Session Timeout" in *Oracle Application Express Administration Guide*.

Restricting Password Reuse

Site administrators can restrict password reuse for all administrator, developer, and end-user accounts based on a history of passwords previously used for the accounts.

To restrict password reuse:

1. Start SQL*Plus and connect to the database where Oracle Application Express is installed having the `apex_administrator_role` role or as SYS specifying the SYSDBA role. For example:

- On Windows:

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

- On UNIX and Linux:

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



Tip:

The `APEX_ADMINISTRATOR_ROLE` must be granted directly to a user and not through another database role.

2. Execute the following statement:

```
ALTER SESSION SET CURRENT_SCHEMA = APEX_180200;
```

3. Execute the following statement:

```
BEGIN
APEX_INSTANCE_ADMIN.SET_PARAMETER('PASSWORD_HISTORY_DAYS',365);
COMMIT;
END;
/
```

This restricts new or updated passwords to those that have not been used for the account for the previous 365 days. Recording of previously used passwords (actually, hashed representations of passwords) begins upon installation of Oracle Application Express.

4. To disable this feature, run the block above, providing 0 for the parameter value. For example:

```
BEGIN
APEX_INSTANCE_ADMIN.SET_PARAMETER('PASSWORD_HISTORY_DAYS',0);
COMMIT;
END;
/
```


 **See Also:**

- `APEX_UTIL.STRONG_PASSWORD_CHECK` procedure in *Oracle Application Express API Reference*
- `APEX_UTIL.STRONG_PASSWORD_VALIDATION` function in *Oracle Application Express API Reference*

About Using SQL in Websheets

Disabling the Allow SQL Worksheet attribute enables administrators to prevent Worksheet users from accessing underlying database objects using SQL tags or from creating SQL reports.

 **See Also:**

"Enabling and Disabling SQL Access in Websheets" in *Oracle Application Express Administration Guide*

About Enabling RESTful Services

Administrators can control the ability to create and access RESTful Services at either the workspace or instance-level.

Disabling the **Enable RESTful Services** attribute prevents developers from creating and editing RESTful Web Services mapped to SQL and PL/SQL in conjunction with Oracle REST Data Services 2.0 or later.

To configure Enable RESTful Services:

- In Workspace Administration — Sign in to Oracle Application Express and click the **Administration** menu. Select **Manage Service, Set Workspace Preferences, SQL Workshop** and find the **Enable RESTful Services** attribute.
- In Instance Administration — Sign in to Oracle Application Express Administration Services. Under Instance Settings, click **Feature Configuration** and click **Feature Configuration**. Find the **REST** tab and then **Enable RESTful Services** attribute.

 **See Also:**

- "Enabling RESTful Services for a Workspace" in *Oracle Application Express Administration Guide*
- "Enabling REST for an Instance" in *Oracle Application Express Administration Guide*

Configuring Instance Settings

Security best practices when configuring an Oracle Application Express instance include determining whether to automate the creation workspaces., enabling and configuring email, configuring storage options, and creating wallets.

- [About Enabling Automation When Creating Workspaces](#)
- [About Enabling and Configuring Email](#)
- [About Configuring Storage Options](#)
- [About Creating Wallets](#)

See Also:

"Configuring Instance Settings" in *Oracle Application Express Administration Guide*

About Enabling Automation When Creating Workspaces

Administrators determine the amount of automation when provisioning (or creating) new workspaces.

To determine how provisioning works, an Instance Administrator selects one of the following options on the Instance Settings page:

- **Manual** - The administrator creates new workspaces and manually notifies the Workspace administrator of the login information.
- **Request** - Users request workspaces directly in a self-service fashion.
- **Request with Email Verification** - In this mode, users request workspaces directly by clicking a link on the login page to access a request form. Each user receives an initial email containing a verification link. When the user clicks the verification link, the request is processed. The user can then access the workspace using the Sign In page.

See Also:

"About Specifying How Workspaces Are Created" in *Oracle Application Express Administration Guide*

About Enabling and Configuring Email

In order to send email from an Oracle Application Express application, an Instance administrator must define the email SMTP settings and set the Use SSL/TLS attribute to use a secure connection for Oracle Database 11g R2 or later.

 **See Also:**

- "Configuring Email" in *Oracle Application Express Administration Guide*
- "[When and Why Network Services Must be Enabled](#)"

About Configuring Storage Options

Administrators can configure the following storage options: require a new schema when requesting a workspace, auto extend tablespaces, or delete uploaded files after a specified number of days.

 **See Also:**

"Configuring Storage" in *Oracle Application Express Administration Guide*

About Creating Wallets

A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL. You must create a wallet if you:

- Call a SSL-enabled URL (for example, by invoking a Web service).
- Create a region of type URL that is SSL-enabled.
- Configure secure SMTP by setting the Use SSL/TLS attribute to **Yes**.
- Have applications with LDAP authentication schemes that are configured to use SSL with Authentication.

 **See Also:**

"Configuring Wallet Information" in *Oracle Application Express Administration Guide*

About Configuring Workspace Purge Settings

Save storage space and improve system performance by purging inactive workspaces.

Inactive workspaces consume valuable storage space and degrade system performance. By enabling Workspace Purge Settings, administrators can configure Oracle Application Express to purge inactive workspaces.

 **See Also:**

"Purging Inactive Workspaces" in *Oracle Application Express Administration Guide*

Understanding Workspace Management

Instance administrators can define associations between workspaces and schemas. Administrators can manage developers and users at the workspace or instance-level and control user access to key components such as App Builder, Team Development, and SQL Workshop, and PL/SQL editing.

- [About Managing Workspace to Schema Assignments](#)
- [About Managing Developers and Users](#)
- [About Managing Component Availability](#)

 **See Also:**

"Managing Existing Workspaces" in *Oracle Application Express Administration Guide*

About Managing Workspace to Schema Assignments

When users log in to Oracle Application Express, they log in to a shared work area called a workspace. Each workspace can have multiple associated (or mapped) schemas. Instance administrators can define associations between workspaces and schemas as appropriate. By associating a workspace with a schema, developers in that workspace can create new database objects in that schema and build applications that interact with the database objects in that schema.

 **See Also:**

"Managing Workspace to Schema Assignments" in *Oracle Application Express Administration Guide*

About Managing Developers and Users

Administrators can define what schemas specific users have access to and also their user role, that is if they are a workspace administrator, a developer, or an end user. Administrators can also limit developer access to specific Application Express components and lock accounts. Oracle Application Express includes two separate interfaces for managing developers and users. Workspace administrators manage user accounts within their workspace and Instance administrators manage all user accounts across an Application Express development instance.

 **See Also:**

- "Managing Users Across an Application Express Instance" in *Oracle Application Express Administration Guide*
- "Managing Users in a Workspace" in *Oracle Application Express Administration Guide*

About Managing Component Availability

Workspace administrators can control user access to key components such as App Builder, Team Development, SQL Workshop, and PL/SQL editing, which determines whether developers have the ability to edit and compile PL/SQL program units from Object Browser. For example, suppose an administrator wants users to be able to build database components, run SQL statements, but not create applications, he or she could define a workspace with rights to a specific schema and then configure the users as developers using the options on the Set Workspace Preferences page.

 **See Also:**

"Configuring Workspace Preferences" in *Oracle Application Express Administration Guide*

About Integrating with Oracle BI Publisher

Oracle recommends that you employ Secure Sockets Layer (SSL) when integrating Oracle Application Express with Oracle BI Publisher. Once SSL (HTTPS protocol) is configured for your Oracle BI Publisher server, you must create a wallet and specify the HTTPS protocol for the Report Server in the internal administration Environment Settings.

 **See Also:**

- "Configuring Wallet Information" in *Oracle Application Express Administration Guide*
- "Configuring Report Printing" in *Oracle Application Express Administration Guide*

About the Advantages of the Application Express Runtime Environment

An Oracle Application Express runtime environment enables you to run production applications, but does not provide a web interface for administration or direct development of these applications

Oracle recommends that you run any sensitive production Oracle Application Express applications with a runtime installation of Oracle Application Express. A runtime installation does not expose the web-based application development environment, thus preventing the use of App Builder, SQL Workshop, and related utilities on a production installation. Additionally, a runtime environment only includes the Oracle Application Express database objects and privileges necessary to run applications, making it a more hardened environment.

See Also:

"Installing Exported Applications into a Runtime Environment" in *Oracle Application Express Administration Guide*

Enabling Network Services in Oracle Database 11g or Later

You must enable network services in Oracle Database 11g or later versions to send outbound mail, use Web services, or use PDF report printing in Oracle Application Express.

- [When and Why Network Services Must be Enabled](#)
Enabling network services enables support for sending outbound mail in Oracle Application Express, use of Web services in Oracle Application Express, and PDF report printing.
- [Granting Connect Privileges Prior to Oracle Database 12c](#)
Demonstrates how to grant connect privileges to any host for the APEX_180200 database user.
- [Granting Connect Privileges in Oracle Database 12c or Later](#)
Procedures CREATE_ACL, ASSIGN_ACL, ADD_PRIVILEGE and CHECK_PRIVILEGE in DBMS_NETWORK_ACL_ADMIN are deprecated in Oracle Database 12c. Oracle recommends to use APPEND_HOST_ACE.
- [Troubleshooting an Invalid ACL Error](#)
Learn how to identify a invalid ACL error by running the query.

When and Why Network Services Must be Enabled

Enabling network services enables support for sending outbound mail in Oracle Application Express, use of Web services in Oracle Application Express, and PDF report printing.

By default, the ability to interact with network services is disabled in Oracle Database 11g Release 2 or later. Therefore, if you are running Oracle Application Express with

Oracle Database 11g Release 2 or later, you must use the new `DBMS_NETWORK_ACL_ADMIN` package to grant connect privileges to any host for the `APEX_180200` database user. Failing to grant these privileges results in issues with:

- Sending outbound mail in Oracle Application Express.
Users can call methods from the `APEX_MAIL` package, but issues arise when sending outbound email.
- Using Web services in Oracle Application Express.
- PDF report printing.

 **Tip:**

To run the examples described in this section, the compatible initialization parameter of the database must be set to at least 11.1.0.0.0. By default a 11g or 12c database will already have the parameter set properly, but a database upgraded to 11g or 12c from a prior version may not. For information about changing database initialization parameters, see "Creating and Configuring an Oracle Database" in [Oracle Database Administrator's Guide](#).

Granting Connect Privileges Prior to Oracle Database 12c

Demonstrates how to grant connect privileges to any host for the `APEX_180200` database user.

The following example demonstrates how to grant connect privileges to any host for the `APEX_180200` database user. This example assumes you connected to the database where Oracle Application Express is installed as `SYS` specifying the `SYSDBA` role.

```
DECLARE
  ACL_PATH VARCHAR2(4000);
BEGIN
  -- Look for the ACL currently assigned to '*' and give APEX_180200
  -- the "connect" privilege if APEX_180200 does not have the privilege
  -- yet.

  SELECT ACL INTO ACL_PATH FROM DBA_NETWORK_ACLS
  WHERE HOST = '*' AND LOWER_PORT IS NULL AND UPPER_PORT IS NULL;

  IF DBMS_NETWORK_ACL_ADMIN.CHECK_PRIVILEGE(ACL_PATH, 'APEX_180200',
  'connect') IS NULL THEN
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(ACL_PATH,
    'APEX_180200', TRUE, 'connect');
  END IF;

EXCEPTION
  -- When no ACL has been assigned to '*'.
  WHEN NO_DATA_FOUND THEN
    DBMS_NETWORK_ACL_ADMIN.CREATE_ACL('power_users.xml',
    'ACL that lets power users to connect to everywhere',
```

```

        'APEX_180200', TRUE, 'connect');
    DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL('power_users.xml','*');
END;
/
COMMIT;

```

The following example demonstrates how to provide less privileged access to local network resources. This example enables access to servers on the local host only, such as email and report servers.

```

DECLARE
    ACL_PATH VARCHAR2(4000);
BEGIN
    -- Look for the ACL currently assigned to 'localhost' and give
    APEX_180200
    -- the "connect" privilege if APEX_180200 does not have the privilege
    yet.
    SELECT ACL INTO ACL_PATH FROM DBA_NETWORK_ACLS
        WHERE HOST = 'localhost' AND LOWER_PORT IS NULL AND UPPER_PORT IS NULL;

    IF DBMS_NETWORK_ACL_ADMIN.CHECK_PRIVILEGE(ACL_PATH, 'APEX_180200',
        'connect') IS NULL THEN
        DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(ACL_PATH,
            'APEX_180200', TRUE, 'connect');
    END IF;

EXCEPTION
    -- When no ACL has been assigned to 'localhost'.
    WHEN NO_DATA_FOUND THEN
        DBMS_NETWORK_ACL_ADMIN.CREATE_ACL('local-access-users.xml',
            'ACL that lets users to connect to localhost',
            'APEX_180200', TRUE, 'connect');
        DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL('local-access-users.xml','localhost');
END;
/
COMMIT;

```

Granting Connect Privileges in Oracle Database 12c or Later

Procedures `CREATE_ACL`, `ASSIGN_ACL`, `ADD_PRIVILEGE` and `CHECK_PRIVILEGE` in `DBMS_NETWORK_ACL_ADMIN` are deprecated in Oracle Database 12c. Oracle recommends to use `APPEND_HOST_ACE`.

The following example demonstrates how to grant connect privileges to any host for the `APEX_180200` database user. This example assumes you connected to the database where Oracle Application Express is installed as `SYS` specifying the `SYSDBA` role.

```

BEGIN
    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
        host => '*',
        ace => xs$ace_type(privilege_list => xs$name_list('connect'),
            principal_name => 'APEX_180200',
            principal_type => xs_acl.ptype_db));

```



```
END;
/
```

The following example demonstrates how to provide less privileged access to local network resources. This example enables access to servers on the local host only, such as email and report servers.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host => 'localhost',
    ace => xs$ace_type(privilege_list => xs$name_list('connect'),
                      principal_name => 'APEX_180200',
                      principal_type => xs_acl.ptype_db));
END;
/
```

Troubleshooting an Invalid ACL Error

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

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

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

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

Next, run the following code to fix the ACL:

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

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

  IF (CNT > 0) THEN

    FOR R IN (SELECT PRINCIPAL FROM XDS_ACE
```

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

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

END;
/

REM commit the changes.

COMMIT;
```

Once the ACL has been fixed, you must run the first script in this section to apply the ACL to the APEX_180200 user.

Understanding Developer Security Best Practices

Developer security best practices include configuring password item types, identifying risky password items, viewing the Security Profiles report, understanding cross site-scripting protection, managing session state, preventing URL tampering, and securing file uploads.

- [About Items of Type Password](#)
Password items do not emit the text entered to the web browser screen. When creating password items, Oracle recommends using password attributes that do not save session state to prevent the password from being saved in the database in the session state tables.
- [Identifying At Risk Password Items](#)
Use the Security Profiles report and Password Items report to help identify at risk password items.
- [Understanding Cross-Site Scripting Protection](#)
Learn about best practices for protecting your application from a cross site-scripting security breach.
- [About Session State and Security](#)
Learn about managing session state and security.
- [Preventing URL Tampering](#)
Session State Protection is a built-in functionality that prevents hackers from tampering with the URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy.
- [About Securing File Uploads](#)
Learn about developer best practices for securing file uploads.

 **See Also:**

- ["Understanding Administrator Security Best Practices"](#)
- ["About Oracle Application Express Administrator Roles"](#)

About Items of Type Password

Password items do not emit the text entered to the web browser screen. When creating password items, Oracle recommends using password attributes that do not save session state to prevent the password from being saved in the database in the session state tables.

Configurable password item type attributes include:

- **Value Required** - If set to **Yes** and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. If set to No, no validation a NULL value is accepted.
- **Submit when Enter pressed** - If set to **Yes**, when the user presses the ENTER key in the field the page is submitted.
- **Does not save state** - If set to **Yes**, the password is not saved into session state. For security reasons you should always set this attribute to Yes. If you set it to No, consider to set the attribute Store value encrypted in session state to Yes.
- **Authorization Scheme** - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.
- **Session State Protection** - You can select the level of session state protection by setting this attribute to Unrestricted or Restricted.
 - Unrestricted means the item may be set by passing the item in a URL or in a form. No checksum is required in the URL.
 - Restricted means the item may not be set from a browser. Use this when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is applicable only to items that cannot be used as data entry items and is always observed, even if Session State Protection is disabled.
- **Store value encrypted in session state** - Session state that is sensitive can be encrypted when stored in Application Express session state management tables. To maintain session state encrypted for this item set the value to Yes. To learn more, see ["About Session State and Security"](#).

 **See Also:**

- ["About Item Types"](#)
- ["Editing Page-Level Items"](#)

Identifying At Risk Password Items

Use the Security Profiles report and Password Items report to help identify at risk password items.

At risk password items are those that either:

- Do not use a password item type that does not save session state.
- Store an unencrypted value in session state .
- [Viewing the Security Profiles Report](#)
- [Viewing the Password Items Report](#)

Viewing the Security Profiles Report

To view the Security Profiles Report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
The App Builder home page appears.
3. Click the **Workspace Utilities** icon.
4. Locate **Cross Application Reports** on the right side of the window.
5. Under Cross Application Reports, click **Security Profiles** report.

This report list the following information about all applications in the current workspace:

- Application
- Name
- Parsing Schema
- Application Level Authorization Scheme
- Authentication
- Authorization Schemes
- Authorization Schemes
- Pages
- Encrypted Items
- At Risk Password Items

Viewing the Password Items Report

To identify at risk password items:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
The App Builder home page appears.
3. Click the **Workspace Utilities** icon.

4. Locate **Cross Application Reports** on the right side of the window.
5. Under Cross Application Reports, click **Password Items**.

The Password Items report shows all of the password items within the application and indicates if they use encryption and whether they save state. Password items that do neither are highlighted as *At Risk*.

 **Tip:**

For pages that contain password items, set the page attribute **Form Auto Complete** to **Off**. Setting that attribute to **Off** prevents the web browser from attempting to auto complete items on the page.

Understanding Cross-Site Scripting Protection

Learn about best practices for protecting your application from a cross site-scripting security breach.

- [What Is Cross Site-scripting Security Breach?](#)
- [Protecting HTML Regions and Other Static Areas](#)
- [About Protecting Dynamic Output](#)
- [About Protecting Report Regions](#)
- [About Protecting Form Items](#)
- [About Restricting Characters Entered on an Item](#)

What Is Cross Site-scripting Security Breach?

Cross site-scripting (also referred to as XSS) is a security breach that takes advantage of dynamically generated web pages. In a XSS attack, a web application is sent a script that activates when it is read by a user's browser. Once activated, these scripts can steal data, even session credentials, and return the information to the attacker.

If malicious code were introduced into an Oracle Application Express application, it could be rendered into HTML regions and other places within the application during normal page rendering. To prevent the introduction of malicious code into session state, the Application Express engine escapes characters in certain cases.

Protecting HTML Regions and Other Static Areas

Learn how to protect HTML regions and other static display areas.

- [About Protecting HTML Regions and Other Static Areas](#)
- [About Safe Item Display Types](#)
- [About the Rules Used to Determine Whether to Escape Values](#)
- [About Using Safe Item Types to Hold and Emit HTML Markup](#)

About Protecting HTML Regions and Other Static Areas

In HTML regions and other static display areas, you can reference session state using the substitution strings. Special substitution strings available within a template are denoted by the number symbol (#), for example, #ABC#. To reference page or application items use the &ITEM. notation.

Examples of static display areas include HTML regions, page headers and footers, region headers and footers, region titles, button labels, help text, form item labels and post-element text, templates, radiogroup (before and after field text), event success messages, event error messages, navigation bar attributes, application static substitution string values, chart labels and legends, breadcrumbs and list framing text, and calendar text, labels, or legends.

Developers can also append an exclamation mark (!) followed by a predefined filter name to a page or application item name or to a report column reference to escape special characters in the substitution value.

See Also:

- ["About Using Substitution Strings"](#)
- ["Controlling Output Escaping in Substitution Strings"](#)

About Safe Item Display Types

When session state is referenced in this way, the value emitted to the page will not have special characters (<, >, &, ") escaped if the referenced item is Display Only with the attribute Save Session State set to No and Escape Special set to No.

If the referenced item has a display type other than Display Only with the attribute Save Session State set to No, the value emitted to the page will have special characters escaped. Although application-level items are also considered to have a safe display type, they do not actually have display properties like form items do.

See Also:

- ["Understanding Page-Level Items"](#)

About the Rules Used to Determine Whether to Escape Values

The Application Express engine uses predefined smart escaping rules to determine if and when to escape values fetched from session state.

The reason for these rules is that items that use the display types listed previously are often for text containing HTML that is intended to be emitted to the browser without being filtered (that is, escaped). The only way this can be made safe is by the enforcement of the rule that these types of items are always escaped on input to the application. For example, if a user passes some text into a safe item using an Oracle

Application Express f?p URL syntax, the Application Express engine escapes special characters when saving the value into session state. This has two intended results:

1. If the value contained no special characters, the value passed in is saved into session state exactly as it was provided.
2. If the value contained special characters, those characters are escaped when the value is saved into session state.

In either situation, the item can now safely be referenced using an `&ITEM.` notation in any HTML region or other static area mentioned previously.

About Using Safe Item Types to Hold and Emit HTML Markup

You can use the safe item types listed previously to hold and emit HTML markup to the browser. For example, suppose you have a requirement to render some text in bold face by referencing a safe page item named `P1_XXX` (using `&P1_XXX.`) The item `P1_XXX` is presumed to contain the following HTML:

```
<b>ABABABAB</b>
```

You can achieve this by using application controls (computations, processes, item source expressions, item default values, and so on) to store values into these safe items. When values are introduced in this way, you ensure the safety of the content. When you use these methods, the Application Express engine does not escape any special characters when saving the values into session state.

Finally, the safety of safe items is ensured by a rule that prevents those items from being posted on a page and submitted to the Application Express engine as part of a page submission.

About Protecting Dynamic Output

Items fetched from session state and rendered using `http.p` or other methods should be explicitly escaped by the code where it is appropriate to do so. For example, suppose a PL/SQL dynamic content region on a page uses the following:

```
http.p(v('SOME_ITEM'));
```

If the value of the item fetched from session state could contain unintended tags or scripts, you might want to use the following in the region:

```
http.p(apex_escape.html(:SOME_ITEM));
```

However, if you are confident that the fetched value is safe for rendering, you do not need to escape the value. As a developer, you must determine when it is appropriate to not escape output.

As a best practice, follow this rule:

- Never emit an item fetched from session state without escaping it unless the item is a safe type.

The reason for this is that as a developer, there is no way you can prevent a hacker from posting a malicious value into a non-safe item. Even if your application does not present these items visibly to ordinary users, be aware that a hacker can mount a XSS attack using your application if you do not follow this rule.

**See Also:**

["About Safe Item Display Types"](#)

About Protecting Report Regions

The Application Express engine escapes data rendered in the body of a report. References to session state in report headings and messages are fetched from session state using the smart escaping rules so that the values of safe item types are not escaped and the values of other item types are escaped.

Oracle Application Express automatically escapes HTML special characters of a report column when the column's **Escape special characters attribute** is set to **Yes**. If you need to render HTML fragments instead of plain column values (for example, for highlighting), instead of concatenating the HTML fragment in the query itself (which prevents you from using Escape special characters), you should use the report column's **HTML Expression** attribute. In the HTML Expression attribute, you can enter static HTML and embed escaped column values with the #COLUMN# notation. The extended column notation gives you control regarding how Oracle Application Express should escape a column value:

- #COLUMN!HTML# - Escapes reserved HTML characters.
- #COLUMN!ATTR# - Escapes reserved characters in a HTML attribute context.
- #COLUMN!JS# - Escapes reserved characters in a JavaScript context.
- #COLUMN!RAW# - Preserves the original item value and does not escape characters.
- #COLUMN!STRIPHTML# - Removes HTML tags from the output and escapes reserved HTML characters.

For example, suppose you have a report based on this query:

```
SELECT
    empno,
    ename,
    NULL DELETE_LINK
FROM emp
```

In this example, all columns are escaped. You could define a HTML Expression on DELETE_LINK as follows:

```
<a href="javascript:if (confirm('Do you really want to delete #ENAME!JS#?'))
doSubmit('DELETE-#EMPNO#');">Delete</a>
```

This example renders a link that asks if you would like to delete an employee and submits a request to delete the row if the user is confirmed. If you had not used #ENAME!JS# but #ENAME#, a name like O'Neill would cause a syntax error and an attacker could exploit the improper escaping for cross-site scripting.

 **See Also:**

- ["Controlling Output Escaping in Substitution Strings"](#)
- ["Editing Interactive Report Column Attributes "](#)

About Protecting Form Items

When form items, including hidden items, obtain their values during the generation of the form page to be sent to the browser, the resulting text is escaped before rendering. Some of the safe item types are exceptions to this rule to support the intended behavior of each display type.

Some item types have the Security attribute **Escape special characters**. Use the Escape special characters attribute to specify if the value should be escaped or not. The default is set to **Yes**. To display HTML code, set this attribute to **No**.

Developers can also append an exclamation mark (!) followed by a predefined filter name to a page or application item name or to a report column reference, to escape special characters in the substitution value.

 **See Also:**

- ["Controlling Output Escaping in Substitution Strings"](#)

About Restricting Characters Entered on an Item

You can limit cross site-scripting and other injection attacks by restricting the characters users can enter on an item. To accomplish this, use the Restricted Characters attribute in the Security region of the Edit Page Item page. Restricted Characters can be saved in session state. Available options include:

- **All Characters Allowed**
No restriction applies.
- **Whitelist for a-Z, 0-9 and space**
Only allow characters a-z, A-Z, 0-9, and space.
- **Blacklist HTML command characters (<>)**
Do not allow reserved HTML characters
- **Blacklist &<>"/;,*|=% and --:**
Do not allow &, <, >, ", /, ;, ", *, |, =, % and "--" (PL/SQL comment).
- **Blacklist &<>"/;,*|=% or -- and new line**
Do not allow &, <, >, ", /, ;, ", *, |, =, %, "--", and new line characters

If you select a restriction, Oracle Application Express displays an error message if a user tries to save data which does not conform to the selected character restriction.

**See Also:**

["Configuring Page Item Security"](#)

About Session State and Security

Learn about managing session state and security.

Oracle Application Express persists session state in database tables. Session state is preserved in database tables because it is more secure to store the session state on the server side than on the client. Because Oracle Application Express applications use the stateless HTTP protocol, an application's session state across multiple page views is preserved in database tables. Not maintaining a synchronous database connection for each Oracle Application Express application user significantly lessens memory and database resource requirements.

Developers can query the session state stored by Oracle Application Express applications using the App Builder and built-in monitoring pages. Developers and administrators can access session state for any application in the workspace to which they are authenticated.

Developer best practices for managing session state include:

- As a standard part of implementing program control flow, clear the session state of unneeded values using clear-cache page processes or clear-cache directives in URLs used to navigate to pages.
- Use password page item types that do not save state. This prevents the entered passwords from being saved in session state tables in the database.
- When sensitive data must persist in a session, it should be saved in Oracle Application Express session state tables in encrypted form. To use this feature, set the page item's Store value encrypted in session state attribute to **Yes**. Encrypted stored values are automatically decrypted when read.

**Note:**

The objective of encrypted session state is to persist session state such that the unencrypted values cannot be read by other Oracle Application Express developers or database administrators who might have access to the Oracle Application Express session state tables, debug output, or database data files. Oracle recommends that database backups be performed using facilities that encrypt data in the backup files.

**Tip:**

You can encrypt item values up to 4000 bytes in length. Attempts to encrypt values longer than 4000 bytes produces an error message.

Preventing URL Tampering

Session State Protection is a built-in functionality that prevents hackers from tampering with the URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy.

- [How Session State Protection Works](#)
- [Enabling Session State Protection](#)
- [Configuring Session State Protection](#)

How Session State Protection Works

When enabled, Session State Protection uses the Page Access Protection attributes and the Session State Protection item attributes with checksums positioned in $f?p=$ URLs to prevent URL tampering and unauthorized access to and alteration of session state. When Session State Protection is disabled, the page and item attributes related to session state protection are ignored and checksums are not included in generated $f?p=$ URLs.

Enabling Session State Protection

You can enable session state protection from either the Edit Security Attributes page or the Session State Protection page.

Enabling Session State Protection is a two-step process. First, you enable the feature. Second, you set page and item security attributes. You can perform these steps using a wizard, or you can set security attributes for pages and items manually on the Session State Protection page.

- [Enabling Session State Protection from Edit Security Attributes](#)
- [Enabling Session State Protection from Session State Protection](#)

Enabling Session State Protection from Edit Security Attributes

To enable Session State Protection from the Edit Security Attributes page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Shared Components** icon.
4. Under Security, click **Security Attributes**.
5. Scroll down to Session State Protection and select **Enabled** from the Session State Protection list.
6. To configure session Session State Protection, click **Manage Session State Protection**.

The Session State Projection page appears.

7. To enable, disable, or configure Session State Protection using a wizard, click **Set Protection**.
8. Otherwise, navigate to the appropriate page or item attributes page and configure the Security attributes.

 **Tip:**

To disable Session State Protection, perform the same steps again, but select **Disabled** instead of **Enabled**. Disabling Session State Protection does not change existing security attribute settings, but those attributes are ignored at runtime.

- [About the Expire Bookmarks Button](#)

About the Expire Bookmarks Button

Enabling Session State Protection affects whether bookmarked links to the current application work. Consider the following rules:

1. Bookmarked links created after Session State Protection is enabled work if the bookmarked link contains a checksum.
2. Bookmarked links created before Session State Protection is enabled do not work if the bookmarked link contains a checksum.
3. Bookmarks that do not contain checksums or contain unnecessary checksums are not affected by Session State Protection.

During page rendering, the Application Express engine uses a hidden application attribute (a checksum salt) during computation and to verify checksums included in `f?p` URLs. When you enable Session State Protection, the Application Express engine includes checksums. You can reset the checksum salt attribute by clicking **Expire Bookmarks** on the Edit Security Attributes page.

 **Tip:**

Be aware that if you click **Expire Bookmarks**, bookmarked URLs used to access your application that contain previously generated checksums will fail.

On the Confirm Invalidate Bookmarks page, you can also configure the Bookmark Checksum Function which is the hash algorithm used for computing bookmark checksums.

Enabling Session State Protection from Session State Protection

To enable Session State Protection:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Security, select **Session State Protection**.

The Session State Protection page appears. Note the current Session State Protection status (Enabled or Disabled) displays at the top of the page.

2. Click the **Set Protection** button.
The Session State Protection wizard appears.
3. Under Select Action, select **Enable** and click **Next**.
Next, determine whether to set security attributes for pages and items.
4. Select **Enable** and click **Next**.
5. Click **Enable Session State Protection**.

 **Tip:**

To disable Session State Protection, perform the same steps, but select **Disable** instead of **Enable**. Disabling Session State Protection does not change existing security attribute settings, but those attributes are ignored at runtime.

Configuring Session State Protection

 **Tip:**

Before you can configure security attributes, you must first enable Session State Protection. See "[Enabling Session State Protection](#)".

- [About Configuring Session State Protection](#)
- [Reviewing Existing Session State Protection Settings](#)
- [Configuring Session State Protection Using a Wizard](#)
- [Configuring Session State Protection for Pages](#)
- [Configuring Session State Protection for Page Items](#)
- [Configuring Session State Protection for Application Items](#)

About Configuring Session State Protection

Once you have enabled Session State Protection, the next step is to configure security attributes. 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.

Reviewing Existing Session State Protection Settings

You can review a summary of Session State Protection settings for pages, items, and application items on the first page of the Session State Protection wizard.

To view summaries of existing Session State Protection settings:

1. Navigate to the Session State Protection page:

- a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Security, select **Session State Protection**.
The Session State Protection page appears.
2. Click **Set Protection**.
 3. Expand the following reports at the bottom of the page:
 - Page Level Session State Protection Summary
 - Page Item Session State Protection Summary
 - Application Item Session State Protection

Configuring Session State Protection Using a Wizard

When you configure Session State Protection using a wizard, you set a value for specific attribute categories. Those selections are then applied to all pages and items within the application.

To configure Session State Protection using a wizard:

1. Navigate to the Session State Protection page:
 - a. On the Workspace home page, click the App Builder icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Security, select **Session State Protection**.
The Session State Protection page appears.
2. Click **Set Protection**.
The Session State Protection wizard appears.
3. Under Select Action, select **Configure** and click **Next**.
4. For Page Access Protection, select one of the following:
 - **Unrestricted** - The page may be requested using a URL with or without session state arguments (Request, Clear Cache, Name/Values).
 - **Arguments Must Have Checksum** - If Request, Clear Cache, or Name/Value arguments appear in the URL, a checksum must also be provided. The checksum type must be compatible with the most stringent Session State Protection attribute of all the items passed as arguments.
 - **No Arguments Allowed** - A URL may be used to request the page but no Request, Clear Cache, or Name/Value arguments are allowed.
 - **No URL Access** - The page may not be accessed using a URL; however, the page may be the target of a Branch to Page branch type, which does not do a URL redirect.
5. For Application Item Protection, select one of the following:
 - **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.

Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).

- **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is also provided. 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.
- **Restricted - May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this option when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is applicable only to items that cannot be used as data entry items and is always observed even if Session State Protection is disabled.

Use this attribute for application items or for page items with any of these Display As types:

- Display as Text (escape special characters, does not save state)
- Display as Text (does not save state)
- Display as Text (based on LOV, does not save state)
- Display as Text (based on PLSQL, does not save state)
- Text Field (Disabled, does not save state)
- Stop and Start HTML Table (Displays label only)

6. For Page Data Entry Item Protection, select one of the following:

- **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.

Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).

- **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.
7. For Page Display-Only Item Protection, select one of the following:
- **Unrestricted** - The item may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).
 - **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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: Session Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.
 - **Checksum Required: User Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum. 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.
 - **Restricted: May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is always observed, even if Session State Protection is disabled.

This attribute may be used with any of these Display As types:

- Display as Text (escape special characters, does not save state)
- Display as Text (does not save state)
- Display as Text (based on LOV, does not save state)
- Display as Text (based on PLSQL, does not save state)
- Text Field (Disabled, does not save state)

- Stop and Start HTML Table (Displays label only)
- 8. Click **Next**.
- 9. Click **Finish**.

Configuring Session State Protection for Pages

To configure Session State Protection for Pages:

1. Navigate to the Session State Protection page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Security, select **Session State Protection**.The Session State Protection page appears.
2. Click the **Page** icon.
3. You can customize the appearance the page using the Search bar at the top of the page.
4. Select a page number.

The Set Page and Item Protection page appears. The following information displays at the top of the page:

- Application ID and name
 - Session State Protection status (Enabled or Disabled)
 - Page Number
 - Page name
5. For Page Access Protection, select one of the following:
 - **Unrestricted** - The page may be requested using a URL with or without session state arguments (Request, Clear Cache, Name/Values).
 - **Arguments Must Have Checksum** - If Request, Clear Cache, or Name/Value arguments appear in the URL, a checksum must also be provided. The checksum type must be compatible with the most stringent Session State Protection attribute of all the items passed as arguments.
 - **No Arguments Allowed** - A URL may be used to request the page but no Request, Clear Cache, or Name/Value arguments are allowed.
 - **No URL Access** - The page may not be accessed using a URL; however, the page may be the target of a Branch to Page branch type, which does not do a URL redirect.
 6. For Item Types, select **Data Entry Items** or **Display-only Items**.

Data Entry items are items that can be altered using forms and include hidden items. Display-Only items are rendered only and are not submitted with the form.
 7. If you select **Data Entry Items**, select a session state protection level for each item:
 - **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.

Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).

- **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.
8. If you select **Display-only Item**, select a session state protection level for each item:
- **Unrestricted** - The item may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).
 - **Restricted: May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is always observed, even if Session State Protection is disabled. This attribute may be used with any of these Display As types:
 - Display as Text (escape special characters, does not save state)
 - Display as Text (does not save state)
 - Display as Text (based on LOV, does not save state)
 - Display as Text (based on PLSQL, does not save state)
 - Text Field (Disabled, does not save state)
 - Stop and Start HTML Table (Displays label only)
 - **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.
9. Click **Apply Changes**.



See Also:

["About the Search Bar"](#)

Configuring Session State Protection for Page Items

To configure Session State Protection for items:

1. Navigate to the Session State Protection page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Security, select **Session State Protection**.
The Session State Protection page appears.
2. Click the **Page Item** icon.
3. You can customize the appearance the page using the Search bar at the top of the page.
4. Select a page number.
The Edit Session State Protection for Page and Items page appears. The following information displays at the top of the page:
 - Application ID and name
 - Session State Protection status (Enabled or Disabled)
 - Page Number
 - Page name
5. For Page Access Protection, select a session state protection level for each item:
 - **Unrestricted** - The page may be requested using a URL with or without session state arguments (Request, Clear Cache, Name/Values).
 - **Arguments Must Have Checksum** - If Request, Clear Cache, or Name/Value arguments appear in the URL, a checksum must also be provided. The

checksum type must be compatible with the most stringent Session State Protection attribute of all the items passed as arguments.

- **No Arguments Allowed** - A URL may be used to request the page but no Request, Clear Cache, or Name/Value arguments are allowed.
 - **No URL Access** - The page may not be accessed using a URL, however the page may be the target of a Branch to Page branch type, which does not do a URL redirect.
6. For Item Types, select **Data Entry Items** or **Display-only Items**.
- Data Entry items are items that can be altered using forms and include hidden items. Display-Only items are rendered only and are not submitted with the form.
7. If you select **Data Entry Items**, select a session state protection level for each item:
- **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).
 - **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.
8. If you select **Display-only Item**, select a session state protection level for each item:
- **Unrestricted** - The item may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).
 - **Restricted: May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This

attribute is always observed, even if Session State Protection is disabled. This attribute may be used with any of these Display As types:

- Display as Text (escape special characters, does not save state)
 - Display as Text (does not save state)
 - Display as Text (based on LOV, does not save state)
 - Display as Text (based on PLSQL, does not save state)
 - Text Field (Disabled, does not save state)
 - Stop and Start HTML Table (Displays label only)
- **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.

9. Click **Apply Changes**.



See Also:

["About the Search Bar"](#)

Configuring Session State Protection for Application Items

To configure Session State Protection for application items:

1. Navigate to the Session State Protection page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Security, select **Session State Protection**.
The Session State Protection page appears.
2. Click the **Application Item** icon.

3. You can customize the appearance the page using the Search bar at the top of the page.
4. Select an application item.
5. Under Security, select one of the following from the Session State Protection list:
 - **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).
 - **Restricted - May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this option when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is only applicable only to items that cannot be used as data entry items and is always observed even if Session State Protection is disabled. This attribute may be used for application items or for page items with any of these Display As types:
 - Display as Text (escape special characters, does not save state)
 - Display as Text (does not save state)
 - Display as Text (based on LOV, does not save state)
 - Display as Text (based on PLSQL, does not save state)
 - Text Field (Disabled, does not save state)
 - Stop and Start HTML Table (Displays label only)
 - **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). 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** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. 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.
6. Click **Apply Changes**.



See Also:

["About the Search Bar"](#)

About Securing File Uploads

Learn about developer best practices for securing file uploads.

Oracle Application Express enables you to easily build an application that can be used to upload files and to access uploaded files. These files are uploaded into a common file storage table. Although the database view `APEX_APPLICATION_FILES` shows those files associated with your database account (or workspace), programmatic access to the common file storage table does not always require authentication, enabling other users to see your uploaded files.:



See Also:

- ["About the Differences Between Page Items and Application Items"](#)
- ["About Item Types"](#) to learn more about creating a File Browse page-level item

Controlling Access to Applications, Pages, and Page Components

Control access to an application, individual pages, or page components by creating an access control list.



Tip:

To create an access control page, the application schema must have `CREATE TABLE`, `CREATE TRIGGER`, and `CREATE SEQUENCE` privileges.

- [Understanding Access Control](#)
Adding the Access Control feature to an application, creates multiple pages and the following components: an Access Control region, access roles, authorization schemes, a build option, and an Application Setting.
- [Creating an Access Control Using the Create Application Wizard](#)
Create an access control list using the Create Application Wizard.
- [Creating an Access Control Using the Create Page Wizard](#)
Create an access control list using the Create Page Wizard.
- [Managing User Access](#)
Manage user access on the Administration page.

- [Configuring Access Control](#)
Configure user access on the Administration page.
- [About Controlling Access for Pages and Page Components](#)
Control access to a specific page or page component by editing the page or component and selecting an authorization schemes.
- [About Removing an Access Control Created with a Wizard](#)
Remove an access control created with a wizard using build options.
- [Managing Roles and User Assignments](#)
Manage application access control roles and user role assignments on the Application Access Control page..

**See Also:**

["Managing Roles and User Assignments"](#)

Understanding Access Control

Adding the Access Control feature to an application, creates multiple pages and the following components: an Access Control region, access roles, authorization schemes, a build option, and an Application Setting.

- [About Adding Access Control](#)
- [About Access Control Authorization Schemes](#)
- [About Configuring Access Control](#)
- [About Exporting an Application with Access Control](#)

About Adding Access Control

Running the Access Control Wizard creates multiple pages and the following components:

- Adds an Access Control region to the Administration page you specify.
- Creates the access roles: Administrator, Contributor, and Reader.
- Creates the authorization schemes: Administration Rights, Contribution Rights, and Reader Rights.

**Note:**

When you add a new Access Control to an existing application, these authorization schemes are only be created if the names do not exist. For example, if `Administration Rights` already exists (case sensitive comparison), the wizard will not recreate. Instead, the Access Control page will re-use the existing authorization scheme.

- Creates the build option, Feature: Access Control.

- Creates the Application Setting, `ACCESS_CONTROL_SCOPE`.

Developers use the access control list to associate the privileges, `view`, `edit`, and `administration`, with application users. Within the final Access Control UI, each privileges correlates to an access role:

- `View` correlates to the `Reader` role.
- `Edit` correlates to the `Contributor` role.
- `Administration` correlates to the `Administrator` role.

See Also:

- ["Attaching an Authorization Scheme to an Application, Page, or Components"](#)
- ["Managing Roles and User Assignments"](#)

About Access Control Authorization Schemes

When you add the Access Control feature to an application, the PL/SQL Body Wizard creates the following authorization schemes:

- `Administration Rights` – This authorization scheme checks if the current user in the application is assigned `ADMINISTRATOR` role.

```
return APEX_ACL.HAS_USER_ROLE (
    p_application_id=>:APP_ID,
    p_user_name => :APP_USER,
    p_role_static_id => 'ADMINISTRATOR');
```

- `Contribution Rights` – This authorization scheme checks if the current user in the application is assigned the `ADMINISTRATOR` role or the `CONTRIBUTOR` role.

```
if apex_acl.has_user_role (
    p_application_id=>:APP_ID,
    p_user_name => :APP_USER,
    p_role_static_id => 'ADMINISTRATOR') or
    apex_acl.has_user_role (
    p_application_id=>:APP_ID,
    p_user_name=> :APP_USER,
    p_role_static_id=> 'CONTRIBUTOR') then
    return true;
else
    return false;
end if;
```

- `Reader Rights` – This authorization scheme returns `TRUE` if the access control is configured to allow any authenticated user access the application. If this behavior

is not allowed, it checks if the current user in the application is assigned to any application role.

```

if nvl(apex_app_setting.get_value(
    p_name => 'ACCESS_CONTROL_SCOPE'),'x') = 'ALL_USERS' then
    -- allow user not in the ACL to access the application
    return true;
else
    -- require user to have at least one role
    return apex_acl.has_user_any_roles (
        p_application_id => :APP_ID,
        p_user_name      => :APP_USER);
end if;

```


 **See Also:**

["Attaching an Authorization Scheme to an Application, Page, or Components"](#)

About Configuring Access Control


Once you add the Access Control feature, you configure it by running the application and accessing the Access Control region on the Administration page.

Administration

Access Control
 Add


All authenticated users can access this application

| | |
|---------------|---|
| Administrator | 1 |
| Contributor | 0 |
| Reader | 0 |



Users

Change access control settings and disable access control



Access Control

Set level of access for authenticated users of this application

The Access Control region lists currently defined access roles and contains two sections: **Users** and **Access Control**.

Users

Click **Users** to add new users, change a user's role, or disable access control by locking an account.



Tip:

You add additional roles and configure role assignments on the Shared Components, Application Access Control page. See "[Managing Roles and User Assignments](#)."

Access Control

Click **Access Control** to specify the behavior when authenticated users access the application.

For **Any authenticated user may access this application**, select one of the following:

- **No** - Select **No** if all users must be defined in the access control list.
- **Yes** - Select **Yes** to enable authenticated users not in the access control list to use the application.



See Also:

- "[Managing User Access](#)"
- "[Configuring Access Control](#)"

About Exporting an Application with Access Control

When you export an application with the Access Control feature, the application roles, Administrator, Contributor, and Reader, are exported. However, the users assigned to these roles are not exported. If you deploy an exported application with the Access Control feature, navigation menu entry for Administration page will not display. When you deploy an application with Access Control feature, you can add user roles as needed by going to Shared Components, Application Access Control. If the application is being deployed in a runtime environment, you can add user roles using APEX_ACL API. For example, the following example adds the user name 'SCOTT' as Administrator in application 255:

```
begin
  APEX_ACL.ADD_USER_ROLE (
    p_application_id => 255,
    p_user_name      => 'SCOTT',
    p_role_static_id => 'ADMINISTRATOR' );
end;
```

You can also execute the `APEX_ACL` API from the command line or create an install script in application supporting objects.

 **See Also:**

- ["Attaching an Authorization Scheme to an Application, Page, or Components"](#)
- ["Managing Roles and User Assignments"](#)
- ["APEX_ACL" in Oracle Application Express API Reference](#)

Creating an Access Control Using the Create Application Wizard

Create an access control list using the Create Application Wizard.

To create an access control list using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Click **Application**.
4. For Name, enter the name used to identify the application to developers.
5. For Appearance, accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
6. To add a new page (such as a blank page, calendar, chart, form, report, and so on) click **Add Page** and select the desired page type.
7. Under Features, select **Access Control**.
8. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
9. Click **Create Application**.

 **See Also::**

- ["Understanding Page Types, Features, and Settings"](#)

Creating an Access Control Using the Create Page Wizard

Create an access control list using the Create Page Wizard.

 **Tip:**

This section describes how to create an access control list by running the Create Page Wizard from the Application home page. You can also run this wizard in Page Designer by clicking the **Create** menu and selecting **Page**.

To create an access control list using the Create Page Wizard:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
The Application home page appears.
3. Click the **Create Page** button.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional). This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined
 - b. Page Type - Select the type of page you want to create.
 - **Component** - Provides page-level functionality and can be added multiple times within a given application. Examples include reports, forms, plug-ins, charts, calendar and so on.
 - **Feature** - Provide application-level functionality and can only be added once per application.

 **Tip:**

The options that appear change based on the selected Page Type. To learn more, "[Understanding Page Types in the Create Page Wizard](#)."

5. For Page Type, select **Feature** and then **Access Control**.
6. On the Create Access Control:
 - a. Administration Page Number - Enter an unused starting page number. This wizard creates multiple pages starting with the number specified.
 - b. Page Group - Identify the name of the page group you to associate with this page. Page groups help developers manage the pages within an application. To create a page group, enter the name. To use an existing page group, select the name from the list.
 - c. Build Option - The wizard creates this build option to support this feature. You can use the build option to control whether the feature appears or to remove it later on.

- d. Administration Page Preference - Specify the administration page to which this access control list will be added. Options include:
 - **Create a new page**
 - **Identify an existing page**What appears next depends upon your selection. Follow the on-screen instructions.
 - e. Click **Next**.
A summary page appears.
7. Click **Create**.

Managing User Access

Manage user access on the Administration page.

Use Manage User Access page to add new users, change a user's role, or disable access control by locking an account.

Tip:

You can also add users by clicking the **Add** button adjacent to the Access Control region heading.

To manage user access:

1. Run the application and navigate to Access Control.
By default, navigate to the **Administration** page and locate **Access Control** region.
2. Under Access Control, click **Users**.
The Manage User Access appears.
3. To add a user:
 - a. Click the **Add User** .
 - b. On Manage User Access, edit the attributes:
 - Username
 - Email
 - Role - Select role (for example, **Administrator**, **Contributor**, or **Reader**).
 - c. Click **Add User**.
4. To edit an existing user:
 - a. Click the **Edit** icon adjacent to the Username.
 - b. On Manage User Access, edit the following:
 - Username
 - Email
 - Role - Select role (for example, **Administrator**, **Contributor**, or **Reader**).

- Account Locked - Select **Yes** or **No**.
 - c. Click **Apply Changes**.
5. To add multiple users:
 - a. Role - Select a role (such as **Administrator**, **Contributor**, or **Reader**).
 - b. Username - Enter usernames separated by commas, semicolons, or white space. Existing or duplicate usernames are automatically be ignored.
 - c. Confirm the additions and click **Apply Changes**.
 6. To delete a user:
 - a. Click the **Edit** icon adjacent to the Username.
 - b. Click **Delete**
 7. To revert you changes, click **Reset**.

Configuring Access Control

Configure user access on the Administration page.

Use Configure Access Control to specify the behavior when authenticated users access the application, determine the default role for users not in the access control list, and determine the userame format (email or string).

To configure access control:

1. Run the application and navigate to Access Control.

By default, navigate to the **Administration** page and locate **Access Control** section.
2. Under Access Control, click **Access Control**.
3. For Any authenticated user may access this application, select either:
 - **No** - Select **No** if all defined users are included in the access control list.
 - **Yes** - Select **Yes** if authenticated users not in the access control list may also use this application.
4. For Treat authenticated users not in ACL as, select either:
 - **Readers** - **Readers** only have `view` privileges.
 - **Contributors** - **Contributors** have `view` and `edit` privileges
5. For Username format, select:
 - **E-mail Address**
 - **String**
6. To save your changes, click **Apply Changes**.

About Controlling Access for Pages and Page Components

Control access to a specific page or page component by editing the page or component and selecting an authorization schemes.

The Access Control Wizard creates authorization schemes that correspond to the application mode list options and the privileges available in the Access Control List.

You can control access to a specific page or page component by editing the page or component and selecting one of the following authorization schemes:

- **Administration Rights** - Only users with Administrator privileges can view the page or component.
- **Contribution Rights** - Users with both Edit and Administrator privileges can view the page or component. Users with View privileges cannot view the page or component.
- **Reader Rights** - Users with Administrator, Edit, or View privileges can view the page or component.
- **Not Administration Rights** - Users with Administrator privileges cannot view the page or component.
- **Not Contribution Rights** - Users with both Edit and Administrator privileges cannot view the page or component. Users with View privileges can view the page or component.
- **Not Reader Rights** - Users with Administrator, Edit, or View privileges cannot view the page or component.



See Also:

["Attaching an Authorization Scheme to an Application, Page, or Components"](#)

About Removing an Access Control Created with a Wizard

Remove an access control created with a wizard using build options.

When you add an access control list using a wizard, the wizard creates one or more pages and other components and processes to seamlessly integrate it into your application. Each feature is associated with a build option which contains one or more components. You use the associated build option to enable, disable, or permanently remove features.

Using Build Options to Include or Exclude Features

Build options have two possible values: **Include** and **Exclude**. If you select the build option status of **Include**, then the Application Express engine considers the associated components (in this case features) as part of the application definition at runtime. Conversely, if you specify the build options status as **Exclude**, then the Application Express engine treats it and any associated components as if it did not exist.

Removing Features Permanently

You remove features created with a wizard by first removing the components associated with the build option and then deleting the build option.

 **See Also:**

- ["Including or Excluding Build Options"](#)
- ["Deleting Build Options and Associated Components"](#)

Managing Roles and User Assignments

Manage application access control roles and user role assignments on the Application Access Control page..

 **Tip:**

You also use the `APEX_APP_ACL` API.

- [About Application Access Control](#)
Learn about managing access control application users and roles.
- [Creating Access Control Roles](#)
Create application access control roles.
- [Editing or Deleting Access Control Roles](#)
Edit or delete application access control roles.
- [Adding User Role Assignments](#)
Define additional user role assignments on the Application Access Control page.
- [Editing User Role Assignments](#)
Edit or delete user role assignments.

 **See Also:**

- ["Controlling Access to Applications, Pages, and Page Components"](#)
- ["APEX_ACL" in Oracle Application Express API Reference](#)

About Application Access Control

Learn about managing access control application users and roles.

You create an access control list by running the Access Control Wizard from either the Create Application Wizard or Create Page Wizard. The Access Control Wizard creates a page to manage an access control list and creates two tables within the application's default parsing schema to manage the access control list. Use the access control list within the application to associate the privileges (`view`, `edit`, and `administration`), with application users. Each privileges correlates to an access level role:

- `View` correlates to the `READER` role.
- `Edit` correlates to the `CONTRIBUTOR` role.

- Administration correlates to the `ADMINISTRATOR` role.

To control access to application pages and components, you need to create an Authorization Scheme and associate it with the application.

About Defining Additional Roles

You can define additional roles on the Application Access Control page. Since roles are applied to users you must create the roles before adding users. Roles and users defined on the Application Access Control page can be reviewed using the following view:

- `APEX_APPL_ACL_USERS`
- `APEX_APPL_ACL_USER_ROLES`
- `APEX_APPL_ACL_ROLES`



See Also:

["Attaching an Authorization Scheme to an Application"](#)

Creating Access Control Roles

Create application access control roles.

Tip:

Since roles are applied to users, you must create the roles before adding users.

To create an application access control role:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Access Control**.
The Application Access Control page appears.
3. Under Roles, click **Add Role**.
The Role dialog appears.
4. On Role:
 - a. Name - Enter a descriptive name for this role. Name may only contain alphanumeric characters and underscores (`_`).
 - b. Static Identifier - Alternate application identifier for this role.

- c. Description - Enter an optional description of this role.
- 5. Click **Create Role**.
The new role displays under Roles on the Application Access Control page.

Editing or Deleting Access Control Roles

Edit or delete application access control roles.

To edit an application access control role:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Access Control**.
The Application Access Control page appears.
3. To edit a role:
 - a. Under Roles, select the role.
The Role dialog appears.
 - b. Edit the attributes.
 - c. Click **Apply Changes**.
4. To delete a role:
 - a. Under Roles, select the role.
The Role dialog appears.
 - b. Click **Delete**.

Adding User Role Assignments

Define additional user role assignments on the Application Access Control page.

To add user role assignments:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Access Control**.
The Application Access Control page appears.
3. Under User Role Assignments, click **Add User Role Assignment**.
The User Assignment dialog appears.
4. On User Assignment:

- a. User Name - Enter a descriptive name for this role. Name may only contain alphanumeric characters and underscores (_).
 - b. Application Role - Select a role.
5. Click **Create Assignment**.

The new user assignment displays under User Role Assignments.

 **Tip:**

Application users are not exported as part of your application. When you deploy your application you will need to manually manage your user to role assignments. Roles are exported as part of an application export and imported with application imports.

Editing User Role Assignments

Edit or delete user role assignments.

To edit user role assignments:

1. Navigate to the Shared Components page:
 - a. On the Workspace home page, click **App Builder**.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
2. Under Application Logic, select **Application Access Control**.
The Application Access Control page appears.
3. To edit an existing user role assignment:
 - a. Under User Role Assignments, select a user name.
The User Assignment dialog appears.
 - b. For Application Role, select a new role.
 - c. Click **Save**.
4. To delete a user role assignment:
 - a. Under User Role Assignments, select a user name.
The User Assignment dialog appears.
 - b. Click **Delete**.

 **Tip:**

Application users are not exported as part of your application. When you deploy your application you will need to manually manage your user to role assignments. Roles are exported as part of an application export and imported with application imports.



See Also:

["Exporting an Application and Application Components"](#)

Establishing User Identity Through Authentication

Use authentication to establish a user's identity to control access to an application. Authentication may require a user identify a user name and password or could involve the use of digital certificates or a secure key.

- [Understanding Authentication](#)
Learn about authentication.
- [Creating an Authentication Scheme](#)
Create an authentication scheme on the Shared Components page.
- [Understanding Preconfigured Authentication Schemes](#)
Learn about the preconfigured authentication schemes.
- [Managing Existing Authentication Schemes](#)
Manage available authentication schemes in the Authentication Schemes Repository.
- [Viewing the Authentication Scheme Associated with an Application](#)
View the current authentication scheme associated with an application.
- [Changing the Authentication Scheme Associated with an Application](#)
Change the authentication scheme associated with an application.
- [Creating a Login Page](#)
Create an application login page.
- [Viewing Authentication Scheme Reports](#)
View authentication scheme reports.

Understanding Authentication

Learn about authentication.

- [How Authentication Works](#)
- [About Support for Deep Linking](#)
- [About Determining Whether to Include Authentication](#)

How Authentication Works

You determine how your application interacts with users. If all users have the same rights and privileges, they are referred to as public users. However, if your application must track each user individually, you must specify an authentication method.

Authentication establishes the identity of each user who accesses your application. Many authentication processes require that a user provide some type of credentials such as a user name and password. These credentials are then evaluated and they either pass or fail. If the credentials pass, the user has access to the application. Otherwise, access is denied.

Once a user has been identified, the Application Express engine keeps track of each user by setting the value of the built-in substitution string `APP_USER`. As a user navigates from page to page, the Application Express engine sets the value of `APP_USER` to identify the user. The Application Express engine uses `APP_USER` as one component of a key for tracking each user's session state.

From a programming perspective, you can access `APP_USER` using the following syntax:

- As a bind variable from either PL/SQL or SQL:

```
:APP_USER
```

- From PL/SQL packages and triggers:

```
V('APP_USER')
```

- As an attribute of the context `APEX$SESSION`:

```
sys_context('APEX$SESSION', 'APP_USER')
```

You can use `APP_USER` to perform your own security checks and conditional processing. For example, suppose you created the following table:

```
CREATE TABLE my_security_table (  
  user_id VARCHAR2(30),  
  privilege VARCHAR2(30));
```

Once created, you could populate this table with user privilege information and then use it to control the display of pages, tabs, navigation bars, buttons, regions, or any other control or component.

See Also:

- ["APP_USER"](#)
- ["Configuring Security Attributes"](#)

About Support for Deep Linking

Oracle Application Express applications that use authentication schemes support deep linking. Deep linking refers to the ability to link to an Oracle Application Express page out of context (for example, from a hyperlink in an email or workflow notification). When you link to a page out of context and the application requires the user be authenticated, the user is taken to the login page. After credentials verification, the Application Express engine automatically displays the page that was referenced in the original link.

About Determining Whether to Include Authentication

As you create your application, you must determine whether to include authentication. You can:

- **Choose to not require authentication.** Oracle Application Express does not check any user credentials. All pages of your application are accessible to all users.

- **Select a built-in authentication scheme.** Create an authentication method based on available preconfigured authentication schemes. Depending on which scheme you choose, you may also have to configure the corresponding components of Oracle 10giAS, Oracle Internet Directory, or other external services.
- **Create custom authentication scheme.** Create a custom authentication method to have complete control over the authentication interface. To implement this approach, you must provide a PL/SQL function the Application Express engine executes before processing each page request. This function's Boolean return value determines whether the Application Express engine processes the page normally or displays a failure page.

 **See Also:**

- ["Understanding Preconfigured Authentication Schemes"](#)
- ["Changing the Authentication Scheme Associated with an Application"](#)
- ["Creating and Editing an Authorization Scheme"](#)

Creating an Authentication Scheme

Create an authentication scheme on the Shared Components page.

To create an authentication scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
The Authentication Schemes page appears.
5. You can customize the appearance the page using the Search bar at the top of the page.
6. To create a new authentication scheme, click **Create**.
7. Specify how the scheme should be created by selecting one of the following:
 - **Based on a pre-configured scheme from the gallery.**
 - **As a copy of an existing authentication scheme.**
8. Follow the on-screen instructions.

 **See Also:**

- ["Understanding Preconfigured Authentication Schemes"](#)
- ["Custom Authentication"](#)
- ["About the Search Bar"](#)

Understanding Preconfigured Authentication Schemes

Learn about the preconfigured authentication schemes.

When you create an authentication scheme from the 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. This section describes all preconfigured authentication schemes that ship with Oracle Application Express.

 **Tip:**

Authentication schemes support plug-ins. To learn more, see ["Implementing Plug-ins."](#)

- [Application Express Accounts](#)
Application Express Accounts are user accounts that are created within and managed in the Oracle Application Express user repository. When you use this method, your application is authenticated against these accounts.
- [Custom Authentication](#)
Creating a Custom Authentication scheme from scratch to have complete control over your authentication interface.
- [Database Accounts](#)
Database Account Credentials authentication utilizes database schema accounts to authenticate users.
- [HTTP Header Variable](#)
Authenticate users externally by storing the username in a HTTP Header variable set by the web server.
- [LDAP Directory Verification](#)
Authenticate a user and password with an authentication request to a LDAP server.
- [No Authentication \(using DAD\)](#)
Adopts the current database user. This approach can be used in combination with a `mod_plsql` Database Access Descriptor (DAD) configuration that uses basic authentication to set the database session user.
- [Open Door Credentials](#)
Enable anyone to access your application using a built-in login page that captures a user name.

- [Oracle Application Server Single Sign-On Server](#)
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.
- [Social Sign-In](#)
Supports authentication with Google, Facebook, and other social network that supports OpenID Connect or OAuth2 standards.

Application Express Accounts

Application Express Accounts are user accounts that are created within and managed in the Oracle Application Express user repository. When you use this method, your application is authenticated against these accounts.

- [About Application Express Accounts](#)
- [Setting Up Application Express Accounts Authentication](#)

About Application Express Accounts

Application Express Accounts is a good solution when:

- You want control of the user account repository.
- User name and password-based approach to security is sufficient.
- You do not need to integrate into a single sign-on framework.

Application Express Accounts is an especially good approach when you must get a group of users up and running on a new application quickly.



See Also:

"Managing Application Express Users" in *Oracle Application Express Administration Guide*

Setting Up Application Express Accounts Authentication

To set up Application Express Accounts:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.

- b. Scheme Type - Select **Application Express Accounts**.
- c. Click **Create Authentication Scheme**.

Custom Authentication

Creating a Custom Authentication scheme from scratch to have complete control over your authentication interface.

- [About Custom Authentication](#)
- [Setting Up Custom Authentication](#)
- [About Session Management Security](#)

About Custom Authentication

Custom Authentication is the best approach for applications when any of the following is true:

- Database authentication or other methods are not adequate.
- You want to develop your own login form and associated methods.
- You want to control security aspects of session management.
- You want to record or audit activity at the user or session level.
- You want to enforce session activity or expiry limits.
- You want to program conditional one-way redirection logic before Oracle Application Express page processing.
- You want to integrate your application with non-Oracle Application Express applications using a common session management framework.
- Your application consists of multiple applications that operate seamlessly (for example, more than one application ID).

Tip:

If you are planning on using the same authentication scheme for multiple applications, consider writing a custom authentication plug-in. See "[Implementing Plug-ins](#)."

Setting Up Custom Authentication

To create a custom authentication scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.

6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **Custom**.
8. Fill in the appropriate fields.

To learn more about a specific field, see field-level Help. If Help is available, the item label changes to include a question mark when you pass your cursor over it.
9. Click **Create Authentication Scheme**.



See Also:

["Viewing Field-Level Help"](#)

About Session Management Security

When running custom authentication, Oracle Application Express attempts to prevent two improper situations:

- Intentional attempts by a user to access session state belonging to someone else. However, users can still type in an arbitrary application session ID into the URL.
- Inadvertent access to a stale session state (probably belonging to the same user from an earlier time). This would commonly result from using bookmarks to application pages.

Oracle Application Express checks that the user identity token set by the custom authentication function matches the user identity recorded when the application session was first created. If the user has not yet been authenticated and the user identity is not yet known, the session state being accessed does not belong to someone else. These checks determine whether the session ID in the request can be used. If not, the Application Express engine redirects back the same page using an appropriate session ID.

Database Accounts

Database Account Credentials authentication utilizes database schema accounts to authenticate users.

- [About Database Account Credentials](#)
- [Setting Up Database Account Credentials](#)

About Database Account Credentials

Database Account Credentials requires that a database user (schema) exist in the local database. This authentication method uses the database account user name and password to authenticate the user. Choose Database Account Credentials if having one database account for each named user of your application is feasible and account maintenance using database tools meets your needs.

Setting Up Database Account Credentials

To set up Database Account Credentials:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **Database Accounts**.
 - c. Click **Create Authentication Scheme**.

HTTP Header Variable

Authenticate users externally by storing the username in a HTTP Header variable set by the web server.

- [About HTTP Header Variable](#)
- [Setting Up HTTP Header Variable](#)

About HTTP Header Variable

HTTP Header Variable supports the use of header variables to identify a user and to create an Application Express user session. Use HTTP Header Variable authentication scheme if your company employs a centralized web authentication solution like Oracle Access Manager which provides single sign-on across applications and technologies. User credential verification is performed by these systems and they pass the user's name to Oracle Application Express using a HTTP header variable such as "REMOTE_USER" (which is the default).

Setting Up HTTP Header Variable

To set up HTTP Header Variable:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.

7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **HTTP Header Variable**.
8. Fill in the appropriate fields.
To learn more about a specific field, see field-level Help.
9. Click **Create Authentication Scheme**.



See Also:

["Viewing Field-Level Help"](#)

LDAP Directory Verification

Authenticate a user and password with an authentication request to a LDAP server.

- [About LDAP Directory Verification](#)
- [Setting Up LDAP Directory Verification](#)

About LDAP Directory Verification

You can configure any authentication scheme that uses a login page to use Lightweight Directory Access Protocol (LDAP) to verify the user name and password submitted on the login page.

App Builder includes wizards and edit pages that explain how to configure this option. These wizards assume that an LDAP directory accessible to your application for this purpose already exists and that it can respond to a `SIMPLE_BIND_S` call for credentials verification. When you create an LDAP Credentials authentication scheme, the wizard requests and saves the LDAP host name, LDAP port, DN string, and determines whether to use SSL, exact DN, and optionally a search filter if not using exact DN. An optional preprocessing function can be specified to adjust formatting of the user name passed to the API.

Setting Up LDAP Directory Verification

To set up LDAP Directory:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:

- a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **LDAP Directory**.
8. Under Settings:
- a. Host - Enter the hostname of your LDAP directory server.
 - b. Port - Enter the port number of your LDAP directory host. The default is 389.
 - c. Use SSL - Choose whether to use SSL to bind to the LDAP directory. If SSL with Authentication is chosen, a wallet must be configured for the Application Express instance.
 - d. Distinguished Name (DN) String - Enter the pattern used to construct the fully qualified distinguished name (DN) string to `DBMS_LDAP.SIMPLE_BIND_S` if using exact DN or the search base if using non-exact DN. Use `%LDAP_USER%` as a placeholder for the username. For example:

Exact DN:

```
cn=%LDAP_USER%,l=amer,dc=yourdomain,dc=com
```

Non-Exact DN (Search Base):

```
Non-Exact DN (Search Base)
```

- e. Use Distinguished Name (DN) - Choose whether the LDAP Distinguished Name (DN) String is exact or non-exact. If non-exact, LDAP Distinguished Name (DN) is the search base and you must supply a Search Filter.
- f. LDAP Username Edit Function - You may provide additional code to be executed to transform the username into a format perfectly suited to the LDAP directory entry or LDAP username. The bind variable `:USERNAME` contains the name the end user specified. For example, the following code calls a function which replaces all periods (.) with underscores (_) in the DN string:

```
return apex_custom_auth.ldap_dnprep(p_username => :USERNAME);
```
- g. Username Escaping - Choose how Oracle Application Express escapes special characters in `%LDAP_USER%`. The correct value for this attribute depends on the used LDAP server and whether a LDAP Username Edit Function already escapes special characters.

To learn more about a specific field, see field-level Help.

9. Click **Create Authentication Scheme**.

 **Note:**

If you choose SSL with Authentication as the SSL mode, a wallet must be set up using the Oracle Application Express instance settings. Additionally, the root certification of the LDAP server must be imported into that wallet as a trusted certificate.

 **See Also:**

- ["Viewing Field-Level Help"](#)
- ["Configuring Wallet Information" in *Oracle Application Express Administration Guide*](#)
- ["Using Oracle Wallet Manager" in *Oracle Database Security Guide*](#)

No Authentication (using DAD)

Adopts the current database user. This approach can be used in combination with a mod_plsql Database Access Descriptor (DAD) configuration that uses basic authentication to set the database session user.

- [About DAD Credentials Verification](#)
- [Setting Up DAD Credentials Verification](#)

About DAD Credentials Verification

DAD credentials verification uses the Oracle database native authentication and user mechanisms to authenticate users using a basic authentication scheme. This authentication scheme gets the user name from the DAD either as the value stored in the DAD configuration or, if the account information is not stored in the DAD configuration, as the user name captured using the basic authentication challenge.

To use DAD credentials verification:

- Each application user must have a user account in the Oracle database.
- You must configure a PL/SQL DAD for basic authentication (without account information).

This results in one user name/password challenge for browser session for your application users. The user identity token is then made available in the `APP_USER` item.

DAD database authentication is useful when you must implement an authentication method that requires minimal setup for a manageable number of users. Ideally these users would have self-managed accounts in the database and your use of this authentication method would be short lived (for example, during the demonstration or prototyping stages of development).

The main drawback of this approach is burdensome account maintenance, especially if users do not administer their own passwords, or if their database accounts exist only to facilitate authentication to your application.

Setting Up DAD Credentials Verification

To set up DAD Credentials Verification:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

The Shared Components page appears.

4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **No Authentication**.
8. Click **Create Authentication Scheme**.

Open Door Credentials

Enable anyone to access your application using a built-in login page that captures a user name.

Open Door Credentials authentication is useful during application development.

- [Setting Up Open Door Credentials](#)

Setting Up Open Door Credentials

To set up Open Door Credentials:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **Open Door Credentials**.
8. Click **Create Authentication Scheme**.

Oracle Application Server Single Sign-On Server

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.

- [About Application Server Single Sign-On Server](#)
- [Setting Up Oracle Application Server Single Sign-On](#)

About Application Server Single Sign-On Server

Oracle Application Express applications can operate as partner applications with Oracle Application Server's Single Sign-On (SSO) infrastructure. You must register your application (or register the Application Express engine) as the partner application by following the Oracle Application Server instructions for registering partner applications and install the Oracle 9iAS SSO Software Developer Kit (SDK).

If you choose this approach, your application will not use an integrated login page. Instead, when a user accesses your application in a new browser session, the Application Express engine redirects to the Single Sign-On login page. After the user is authenticated by SSO, the SSO components redirect back to your application, passing the user identity and other information to the Application Express engine. The user can then continue to use the application until they log off, terminate their browser session, or until some other session-terminating event occurs.

Setting Up Oracle Application Server Single Sign-On

To set up Oracle Application Server Single Sign-On:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **Oracle Application Server Single Sign-On**.
8. Fill in the appropriate fields.
To learn more about a specific field, see field-level Help.
9. Click **Create Authentication Scheme**.



See Also:

["Viewing Field-Level Help"](#)

Social Sign-In

Supports authentication with Google, Facebook, and other social network that supports OpenID Connect or OAuth2 standards.

- [About Social Sign-In](#)

- [Setting Up Social Sign-In Authentication](#)

About Social Sign-In

Social Sign-In authentication is primarily useful for the following use cases:

- Your application is internet facing and you expect an unknown number of users from social networks to use your application.
- Your company has standardized on one of these providers, Oracle Identity Cloud Service, an internal OpenID Connect or OAuth2 system for authentication.

User credential verification is performed by these systems. Be aware that anyone who is registered at this provider can use your application, unless you use authorization schemes for protection.

Social network authentication providers store a multitude of information about users. You can configure the Social Login authentication scheme to request this information, using the Scope attribute (see Item Help). In the authentication scheme's Post-Authentication procedure, you can access this information using `APEX_JSON.GET_%` function calls.

You can use multiple authentication schemes in your application, to support more than one Social Login provider and other schemes.

See Also:

"[Authorization](#)" for the authentication scheme attribute `Switch in Session` for details.

Setting Up Social Sign-In Authentication

Note:

As a prerequisite for creating the Social Sign-In Authentication , you have to request OAuth2 credentials from the authentication provider and configure your account to support redirects to your application. Check your authentication provider's documentation for the details. The steps below assume that this has been done and that you saved the OAuth2 credentials in the application's shared components.

To set up Social Sign-In Authentication:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.

5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
 - a. Name - Enter the name used to reference the authentication scheme by other application developers.
 - b. Scheme Type - Select **Social Sign-In**.
8. Fill in the appropriate fields.
To learn more about a specific field, see field-level Help.
9. Click **Create Authentication Scheme**.

Managing Existing Authentication Schemes

Manage available authentication schemes in the Authentication Schemes Repository.

Once created, available authentication schemes display in the Authentication Schemes Repository.

To navigate to the Authentication Schemes Repository:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. You can customize the appearance the page using the Search bar at the top of the page.
6. To edit a specific authentication scheme as a report, click the authentication scheme name.
The Authentication Scheme page appears.
To learn more about a specific field, see field-level Help.
7. Edit the appropriate attributes and click **Apply Changes**.

See Also:

- ["About the Search Bar"](#)
- ["Viewing Field-Level Help"](#)

Viewing the Authentication Scheme Associated with an Application

View the current authentication scheme associated with an application.

To view the current authentication scheme for an application:

1. On the Workspace home page, click the **App Builder** icon.

2. Select an application.
3. Click **Shared Components**.
4. Under Security, click **Security Attributes**.
5. Locate the Authentication section. The current authentication scheme displays next to **Authentication Scheme**.
6. To edit the Authentication Scheme, **Define Authentication Schemes**.

Changing the Authentication Scheme Associated with an Application

Change the authentication scheme associated with an application.

To change the authentication scheme for an application:

1. Navigate to the Authentication Schemes:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. On the Application home page, click **Shared Components**.
The Shared Components page appears.
 - d. Under Security, select **Authentication Schemes**.
2. Select the authentication scheme you want to activate.
The Create/Edit page appears.
3. Click **Make Current Scheme**.



See Also:

["Understanding Preconfigured Authentication Schemes"](#)

Creating a Login Page

Create an application login page.

When you create an application in Oracle Application Express, a login page is created. The alias for the page is 'LOGIN'. You can use this page as the "invalid session page" in an authentication scheme. The page is constructed with processes that call the Oracle Application Express login API to perform credentials verification and session registration.

You can also build your own login pages using the pre-built pages as models and tailoring all of the user interface and processing logic to your requirements.

To create a login page for your application:

1. On the Workspace home page, click the App Builder icon.
2. Select an application.
3. Click **Create Page**.

4. For Create a Page:
 - a. User Interface - Select a user interface for the page.
 - b. Select a page type - Select **Login Page**.
 - c. Click **Next**.
5. Select **Login Page**.
6. Specify Login page attributes and click **Create**.

Viewing Authentication Scheme Reports

View authentication scheme reports.

To view authentication scheme reports:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. Click the appropriate tab at the top of the page:
 - **Subscription** - Use the Authentication Scheme Subscription report to view subscribed authentication schemes in your application.
 - **History** - Use the Authentication Scheme History report to view recent modifications made to items in this application.

Providing Security Through Authorization

Extend the security of your application by creating an authorization scheme.

Authorization is a broad term for controlling access to resources based on user privileges. While conditions control the rendering and processing of specific page controls or components, authorization schemes control user access to specific controls or components.

- [How Authorization Schemes Work](#)
Learn about how an authorization scheme extends the security of your application's authentication scheme.
- [Creating and Editing an Authorization Scheme](#)
Learn how to create and edit an authorization scheme. Before you can attach an authorization scheme to an application or an application component or control, you must first create it.
- [Attaching an Authorization Scheme to an Application, Page, or Components](#)
- [Viewing Authorization Reports](#)

How Authorization Schemes Work

Learn about how an authorization scheme extends the security of your application's authentication scheme.

You can specify an authorization scheme for an entire application, page, or specific control such as a region, item, or button. For example, you could use an authorization scheme to selectively determine which tabs, regions, or navigation bars a user sees. An authorization scheme either succeeds or fails. Common authorization scheme types include Exists, Not Exists SQL Queries, and PL/SQL Function Returning Boolean. If a component or control level authorization scheme succeeds, the user can view the component or control. If it fails, the user cannot view the component or control. If an application or page-level authorization scheme fails, then Oracle Application Express displays a previously defined message.

When you define an authorization scheme, you give it a unique name. Once defined, you can attach it to any component or control in your application. To attach an authorization scheme to a component or control in your application, simply navigate to the appropriate attributes page and select an authorization scheme from the Authorization Scheme list.

Creating and Editing an Authorization Scheme

Learn how to create and edit an authorization scheme. Before you can attach an authorization scheme to an application or an application component or control, you must first create it.

- [About Authorization Scheme Types](#)
- [Creating an Authorization Scheme](#)
Create an authorization scheme on the Shared Components page.
- [Editing Attributes of an Existing Authorization Scheme](#)
Edit attributes of an existing authorization scheme.
- [Changing the Evaluation Point Attribute](#)
Control when an authorization scheme is validated using the Evaluation Point, Validate authorization scheme attribute.
- [About Resetting Authorization Scheme State](#)
Call an API to reset a session's authorization scheme state.

About Authorization Scheme Types

When you create an authorization scheme you select an authorization scheme type. The authorization scheme type determines how an authorization scheme is applied. Developers can create new authorization type plug-ins to extend this list.

Table 20-1 Authorization Scheme Types

| Authorization Scheme Types | Description |
|-----------------------------------|---|
| Exists SQL Query | Enter a query that causes the authorization scheme to pass if it returns at least one row and causes the scheme to fail if it returns no rows |
| NOT Exists SQL Query | Enter a query that causes the authorization scheme to pass if it returns no rows and causes the scheme to fail if it returns one or more rows |
| PL/SQL Function Returning Boolean | Enter a function body. If the function returns true, the authorization succeeds. |
| Item in Expression 1 is NULL | Enter an item name. If the item is null, the authorization succeeds. |
| Item in Expression1 is NOT NULL | Enter an item name. If the item is not null, the authorization succeeds. |

Table 20-1 (Cont.) Authorization Scheme Types

| Authorization Scheme Types | Description |
|---|---|
| Value of Item in Expression 1 Equals Expression 2 | Enter an item name and value. The authorization succeeds if the item's value equals the authorization value. |
| Value of Item in Expression 1 Does NOT Equal Expression 2 | Enter an item name and a value. The authorization succeeds if the item's value is not equal to the authorization value. |
| Value of Preference in Expression 1 Does NOT Equal Expression 2 | Enter a preference name and a value. The authorization succeeds if the preference's value is not equal to the authorization value. |
| Value of Preference in Expression 1 Equals Expression 2 | Enter a preference name and a value. The authorization succeeds if the preference's value equal the authorization value. |
| Is In Group | Enter a group name. The authorization succeeds if the group is enabled as a dynamic group for the session. See "APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS" in <i>Oracle Application Express API Reference</i> . If the application uses Application Express Accounts Authentication, this check also includes workspace groups that are granted to the user. If the application uses Database Authentication, this check also includes database roles that are granted to the user. |
| Is Not In Group | Enter a group name. The authorization succeeds if the group is not enabled as a dynamic group for the session. |

Creating an Authorization Scheme

Create an authorization scheme on the Shared Components page.

To create an authorization scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select Authorization Schemes.
5. Click **Create**.
6. Specify how to create an authorization scheme by selecting one of the following:
 - **From Scratch**
 - **As a Copy of an Existing Authorization Scheme**
7. On Create Authorization Scheme - Details:
 - a. Name - Enter an unique name that identifies this authorization scheme.
 - b. Scheme Type - Select how this authorization scheme will be applied. See "[About Authorization Scheme Types](#)."
 - c. Identify error message displayed when scheme violated - Enter error text that displays if the authorization scheme fails (that is, the current user fails the security check).

- d. Validate Authorization Scheme - Authorization schemes are evaluated on first use in a session. Use this option to controls if future uses cause re-evaluations and when a memorized result can be taken instead.

For more details, see field-level Help.

8. Click **Create Authorization Scheme**.

See Also:

- ["Attaching an Authorization Scheme to an Application, Page, or Components"](#)
- ["Changing the Evaluation Point Attribute"](#)
- ["About Resetting Authorization Scheme State"](#)

Editing Attributes of an Existing Authorization Scheme

Edit attributes of an existing authorization scheme.

To edit attributes of an existing authorization scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

The Shared Components page appears.

4. Under Security, select **Authorization Schemes**.

The Authorization Schemes page appears. By default, each scheme displays as an icon. You can customize the appearance the page using the Search bar at the top of the page.

5. Select an authorization scheme.
6. Edit the appropriate attributes.

To learn more, see field-level Help.

7. To save your changes, click **Apply Changes**.

Changing the Evaluation Point Attribute

Control when an authorization scheme is validated using the Evaluation Point, Validate authorization scheme attribute.

To change the authorization scheme evaluation point:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

The Shared Components page appears.

4. Under Security, select **Authorization Schemes**.

The Authorization Schemes page appears.

By default, each scheme displays as an icon. You can customize the appearance the page using the Search bar at the top of the page.

5. Select an authorization scheme.
6. Scroll down to Evaluation Point and edit the Validate authorization scheme attribute. Options include:
 - **Once per session** - Evaluate only once and always use the memorized result afterwards.
 - **Once per page view** - Evaluate once for each request that is processed. Use the memorized result if the authorization scheme is referenced in more than one component on the page.
 - **Once per component** - Evaluate once for each component that references the authorization scheme, but save the result in the session. Further requests to render or process the scheme use the result that was memorized for the component.
 - **Always (No Caching)** - The authorization scheme will always be evaluated.To learn more, see field-level Help.
7. To save your changes, click **Apply Changes**.

 **Tip:**

The default value **Once per session** is the most efficient. You should choose another value if the authorization check depends on changing session state or other factors that are not consistent over an entire session.

About Resetting Authorization Scheme State

Call an API to reset a session's authorization scheme state.

If an authorization scheme is validated once for each session, Oracle Application Express caches the validation results in each user's session cache. You can reset a session's authorization scheme state by calling the `APEX_AUTHORIZATION.RESET_CACHE` API.

 **See Also:**

"`APEX_AUTHORIZATION.RESET_CACHE` Procedure" in *Oracle Application Express API Reference*

Attaching an Authorization Scheme to an Application, Page, or Components

Once you have created an authorization scheme you can attach it to an entire application, page, control, or component.

- [Attaching an Authorization Scheme to an Application](#)
- [Attaching an Authorization Scheme to a Page](#)
- [Attaching an Authorization Scheme to a Control or Component](#)

Attaching an Authorization Scheme to an Application

To attach an authorization scheme to an application:

1. On the Workspace home page, click the App Builder icon.
2. Select an application.
3. Click the **Shared Components** icon.
The Shared Components page appears.
4. Under Security, click **Security Attributes**.
5. Scroll down to Authorization and make a selection from the Authorization Scheme list.
6. For Run on Public Pages, select Yes or No to specify whether the application-level authorization scheme is checked on public pages (that is, pages that do not require authorization). To learn more, see "[Authorization](#)."
7. To define a new authorization scheme, click **Define Authorization Schemes**.

Attaching an Authorization Scheme to a Page

To attach an authorization scheme to a page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Select a page.
Page Designer appears.
4. In the Rendering tab on the left side, click the page title at the top of the tree.
5. In the Property Editor, locate Security and make a selection from the Authorization Scheme attribute.

Tip:

To find a group or attribute:

- **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
- **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

6. Click **Save**.

Attaching an Authorization Scheme to a Control or Component

To attach an authorization scheme to a page component or control:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Select a page.
Page Designer appears.
4. Select the component or control to which you want to apply the authorization scheme.
5. In the Property Editor, locate Security and make a selection from the Authorization Scheme attribute.

Tip:

To find a group or attribute:

- **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
- **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

6. Click **Save**.

Viewing Authorization Reports

You can use the Authorization Scheme Subscription and Authorization Scheme Utilization reports to better manage authorization schemes within your application.

To view authorization scheme reports:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
The Shared Components page appears.
4. Under Security, select **Authorization Schemes**.
5. Click the appropriate tab at the top of the page:
 - **Subscription** - Use the Authorization Scheme Subscription report to view details about authorization schemes subscription.
 - **By Component** - Use the By Component report to view all components within this application which have an associated authorization scheme. For a component to be rendered it must pass authorization schemes placed on the component level, the page level, and at the application level.

- **Utilization** - Use the Authorization Scheme Utilization report to view details about authorization schemes utilization.

To view additional reports indicating which pages having authorization schemes and which do not, select one of the following from the Tasks list:

- **Report Pages With Authorization Schemes**
- **Report Pages Without Authorization Schemes**
- **History** - Use the Authorization Scheme History report to view recent modifications made to Authorization Schemes in this application.

21

Managing Application Globalization

You can run applications built in App Builder concurrently in different languages.

This section describes how to translate an application built in App Builder.

- [Understanding Application Translation and Globalization Support](#)
A single Oracle database instance and Oracle Application Express can support multiple database sessions customized to support different languages.
- [Specifying the Primary Language for an Application](#)
Globalization attributes specify how the Application Express engine determines the primary language of an application.
- [About Apply Format Masks to Items](#)
The Application Express engine applies globalization settings for each rendered page. This default behavior can impact the display of certain items such as numbers and dates.
- [About Translating Applications for Multibyte Languages](#)
If your application must run in several languages simultaneously (such as Chinese and Japanese), consider configuring your database with a character set to support all of the languages.
- [Understanding the Translation Process](#)
To translate an application developed in App Builder, you must map the primary and target language, seed and export text to a translation file, translate the text, apply the translation file, and publish the translated application.
- [Translating Messages](#)
Learn when and how to translate messages in an Oracle Application Express application.
- [Translating Data That Supports List of Values](#)
You create a dynamic translation to translate dynamic pieces of data. For example, you might use a dynamic translation on a list of values based on a database query.
- [Understanding Supported Globalization Codes](#)
If you are building a multilingual application, it is important to understand how globalization codes affect the way in which your application runs. These codes are set automatically based on the application-level Globalization attributes you select.



See Also:

"Viewing Installed Translations" in *Oracle Application Express Administration Guide*

About Apply Format Masks to Items

The Application Express engine applies globalization settings for each rendered page. This default behavior can impact the display of certain items such as numbers and dates.

For example, suppose your application determines the application language based on the user's browser language preference. If the Application Express engine determines the user's browser language preference is French, it displays dates and numbers in a format that conforms to French standards. You can override this default behavior and explicitly control how items display by applying a format mask. You apply a format mask by making a selection from the Display As list:

- When you create the item.
- After you create the item by editing the item attributes.

Understanding Application Translation and Globalization Support

A single Oracle database instance and Oracle Application Express can support multiple database sessions customized to support different languages.

You can develop applications in App Builder that can run concurrently in different languages.

- [About the Translation Process](#)
Learn about translating an application built in App Builder.
- [About Language Identification](#)
After you create an application, you specify a language preference. Then you select a primary application language and determine how the Application Express engine determines the application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item.
- [About the Rules for Translating Database Applications](#)
Learn about rules for translating database applications.
- [How Translated Applications Are Rendered](#)
After Oracle Application Express determines the language for an application, the Application Express engine alters the database language for a specific page request. It then looks for a translated application in the appropriate language. If the Application Express engine finds that language, it renders the application using that definition. Otherwise, it renders the application in the base (or primary) application language.
- [Making Application Attributes Translatable](#)
Learn about best practices to make application attributes within your application translatable.

About the Translation Process

Learn about translating an application built in App Builder.

In general, translating an application built in App Builder involves the following steps:

1. Map the primary and target application IDs.
2. Seed and export the text to a file for translation.
3. Translate the text in the file.
4. Apply the translated file.
5. Publish the translated file.



See Also:

["Understanding the Translation Process"](#)

About Language Identification

After you create an application, you specify a language preference. Then you select a primary application language and determine how the Application Express engine determines the application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item.



See Also:

["Specifying the Primary Language for an Application"](#)

About the Rules for Translating Database Applications

Learn about rules for translating database applications.

Use the following rules to determine which translated version to use:

- Look for an exact match between the user language preference and the language code of the translated application.
- Look for a truncated match. That is, see if the language and locale exist. For example, if the user language preference is `en-us` and the translated version of `en-us` does not exist, look for a translated application that has the language code `en`.
- Use the primary application language.

For example, suppose you create an application with the primary language of German, `de`, and you create a translated version of the application with a language code of `en-us`. Users accessing this application with a browser language of `en-us` execute the English `en-us` version of the application. Users accessing the application with a browser language of `en-gb` view the application in the application's primary language, that is, in German. For this example, you should create the translated English version using language code `en` to encompass all variations of `en`.

How Translated Applications Are Rendered

After Oracle Application Express determines the language for an application, the Application Express engine alters the database language for a specific page request. It then looks for a translated application in the appropriate language. If the Application Express engine finds that language, it renders the application using that definition. Otherwise, it renders the application in the base (or primary) application language.

Note that the text that displays within an application is not translated on the fly. Oracle Application Express dynamically collects page attributes from either a base language application definition or an alternative application definition.

See Also:

- ["About Dynamic Translation Text Strings"](#)
- ["Translating Data That Supports List of Values"](#)

Making Application Attributes Translatable

Learn about best practices to make application attributes within your application translatable.

When you build an application in App Builder, you define a large number of declarative attributes such as field labels, region headings, page header text, and so on.

- [About Shortcuts that Support Translatable Messages](#)
- [About Messages](#)
- [About Dynamic Translation Text Strings](#)
- [About Translating Region Titles](#)
- [About Translating Templates](#)

About Shortcuts that Support Translatable Messages


App Builder includes two shortcut types that enable you to reference translatable messages:

- **Message.** Use this shortcut to reference a translatable message at runtime. Note that the name of the shortcut must match the corresponding message name. At runtime, the name of the shortcut expands to the text of the translatable message for the current language.
- **Message with JavaScript Escaped Single Quotes.** Use this shortcut to reference a shortcut inside of a JavaScript literal string and reference a translatable message at runtime. This shortcut defines a text string. When the shortcut is referenced, it escapes the single quotation marks required for JavaScript.

 **See Also:**
["Using Shortcuts"](#)

About Messages

If your application includes PL/SQL regions or PL/SQL processes, you must translate any generated HTML or text. You may also need to translate messages used in reports if your application uses a language that is not one of the ten languages into which Oracle Application Express is translated.

 **See Also:**
["Translating Messages"](#)

About Dynamic Translation Text Strings

Dynamic translations are used for database data that must be translated at runtime. For example, you might use a dynamic translation to translate a list of values based on a database query. A dynamic translation consists of a translate-from language string, a language code, and a translate-to string. You can also use the `APEX_LANG.LANG` API to retrieve dynamic translations programmatically.

 **See Also:**
["Translating Data That Supports List of Values"](#)

About Translating Region Titles

By default, page region titles are included in the generated translation file. However, you can mark a region title as not translatable.

- [Marking a Region as Not Translatable in Page Designer](#)

Marking a Region as Not Translatable in Page Designer

To mark a region title as not translatable:

1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
Page Designer appears.

2. In the Rendering tab, select the region.
The Property Editor displays the region attributes. Attributes are organized in groups.
3. To find a group or attribute:
 - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
 - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Expand the Advanced section.
5. For Exclude Title from Translation, select Yes.
6. Click **Save**.

About Translating Templates

By default, templates are not translatable, and therefore are not included in the generated translation file. Generally, templates do not and should not contain translatable text. However, if you must mark a template as translatable, select the Translatable check box on the Edit Page Template page.

To identify a template as translatable:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under User Interface, select **Templates**.
The Templates page appears.
5. Locate the template you want to edit and select the template name.
6. Under Name, select **Translatable**.

You can include translatable text at the application-level by defining the translatable text using static substitution strings. Because application-level attributes are translated, any text defined as a static substitution string is included in the generated translation file.

See Also:

- ["Creating Custom Themes"](#)
- ["Substitutions"](#)

Specifying the Primary Language for an Application

Globalization attributes specify how the Application Express engine determines the primary language of an application.

To edit globalization attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under Globalization, select **Globalization Attributes**.
5. For **Application Primary Language**, select the language in which the application is being developed.
6. For **Application Language Derived From**, specify how the Application Express engine determines (or derives) the application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item. [Table 21-1](#) describes available options:

Table 21-1 Application Language Derived From Options

| Option | Description |
|--|---|
| No NLS (Application not translated) | Select this option if the application will not be translated. |
| Application Primary Language | Determines the application's primary language based on the Application Primary Language attribute. (See step 5.) |
| Browser (use browser language preference) | Determines the application's primary language based on the user's browser language preference. |
| Application Preference (use FSP_LANGUAGE_PREFERENCE) | Determines the application's primary language based on a value defined using the <code>APEX_UTIL.SET_PREFERENCE</code> API. Select this option to maintain the selected language preference across multiple logins. See Also: "SET_PREFERENCE Procedure" in <i>Oracle Application Express API Reference</i> |
| Item Preference (use item containing preference) | Determines the application's primary language based on an application-level item called <code>FSP_LANGUAGE_PREFERENCE</code> . Using this option requires Oracle Application Express to determine the appropriate language preference every time the user logs in. |
| Session | Determines the translated application language from the session setting. The Application Express session language can be set through either the <code>APEX_UTIL.SET_SESSION_LANG</code> procedure or by using the <code>p_lang</code> parameter of the "f" procedure in the URL. |

7. Configure other options as appropriate. To learn more about an attribute, see field-level Help.

8. Click **Apply Changes**.

 **See Also:**

- ["Editing Application Attributes"](#)
- ["Configuring Security Attributes"](#)
- ["Understanding Supported Globalization Codes"](#)

About Translating Applications for Multibyte Languages

If your application must run in several languages simultaneously (such as Chinese and Japanese), consider configuring your database with a character set to support all of the languages.

UTF8 and AL32UTF8 are the character sets you can use to support almost all languages around the world.

Understanding the Translation Process

To translate an application developed in App Builder, you must map the primary and target language, seed and export text to a translation file, translate the text, apply the translation file, and publish the translated application.

This section describes the process of translating an application.

- [Step 1: Define Application Language](#)
The first step in translating an application is to map the primary and target application language. The primary application is the application to be translated. The target application is the resulting translated application.
- [Step 2: Seed and Download to a Translation File](#)
Seed the translation table and then export the translation text to a translation file.
- [Step 3: Translate the XLIFF File](#)
After you export a translatable file to XLIFF format, you can translate it into the appropriate languages.
- [Step 4: Upload and Apply a Translated XLIFF Document](#)
After your XLIFF document has been translated, the next step is to upload and then apply it.
- [Step 5: Publish the Application](#)
Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternate languages.
- [Manually Editing a Translation](#)
Once you have mapped the target language and seeded the translatable text, you manually edit a translation.

 **See Also:**

- ["Translating Messages"](#)
- ["Translating Data That Supports List of Values"](#)

Step 1: Define Application Language

The first step in translating an application is to map the primary and target application language. The primary application is the application to be translated. The target application is the resulting translated application.

To map the primary and target application language:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.The Translate page appears.
2. Click **Define application languages**.
The Mappings page appears.
3. Click **Create**.
4. On the Create/Edit Language Application Mapping:
 - **Translation Application** - Enter a unique integer value that identifies an application. The application ID must be unique on your instance of Oracle Application Express.
 - **Language** - Select the language to be translated.
 - **Image Directory** - Enter the directory where images are obtained. Unless you are using static files from the images directory which are translated, leave this field blank. This is the default and most common option.
5. Click **Create**.

 **See Also:**

["Step 2: Seed and Download to a Translation File"](#)

Step 2: Seed and Download to a Translation File

Seed the translation table and then export the translation text to a translation file.

- [Seeding Translatable Text](#)
- [Downloading an XLIFF File](#)

Seeding Translatable Text

Seeding the translation copies all translatable text into the Translation Text repository. After you specify the language and seed the Translation Text, you can then generate and export an XLIFF file for translation.

The seeding process keeps your primary language application synchronized with the Translation Text repository. You should run the seed process any time your primary language application changes.

To seed translatable text:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.

The Translate page appears.

2. Click **Seed translatable text**.
3. Select the appropriate target application.
4. Click **Seed**.

When completed, the following message displays:

```
Successfully seeded translatable text.
```

Downloading an XLIFF File

Once the translation mappings are established the translatable text within the application is seeded into a translation repository. This repository is then exported to an XML Localization Interchange File Format (XLIFF) file for translation.

- [About the XLIFF Export Page](#)
- [Downloading a XLIFF file for an Application](#)
- [Downloading an XLIFF File for an Application Page](#)
- [About Including XLIFF Target Elements](#)

About the XLIFF Export Page

The XLIFF Export page is divided into two sections. Use the upper section to export translatable text for an entire application (that is, all pages, lists of values, messages, and so on). Use the lower section to export translatable text for a specific page.

 **Note:**

XML Localization Interchange File Format (XLIFF) is an XML-based format for exchanging localization data. For more information about XLIFF and the XLIFF 1.0 specification, see:

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

Downloading a XLIFF file for an Application

To download an XLIFF file for an complete application:

1. Seed the translatable text.
2. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.
The Translate page appears.
3. On the Translate page, click **Download XLIFF translation files**.
4. Under Download XLIFF file for complete Application:
 - a. **Language** - Choose the language to be downloaded in the XLIFF file.
 - b. **Include XLIFF Target Elements** - Specify whether to include XLIFF target elements. To include then, select **Include XLIFF Target Elements**.
 - c. **Export** - Choose to export all translatable elements of the application or only those elements which are new or have been updated.
 - d. Click **Export XLIFF**.
5. Follow the on-screen instructions.

 **See Also:**

["Seeding Translatable Text"](#)

Downloading an XLIFF File for an Application Page

To download an XLIFF file for a specific page:

1. Seed the translatable text.
2. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.

- d. Under Globalization, click **Translate Application**.
The Translate page appears.
3. On the Translate page, click **Download XLIFF translation files**.
4. Under Download XLIFF file for Application Page:
 - a. **Language** - Choose the language to be downloaded in the XLIFF file.
 - b. **Page** - Select a page.
 - c. **Include XLIFF Target Elements** - Specify whether to include XLIFF target elements. To include them, select **Include XLIFF Target Elements**.
 - d. **Export** - Choose to export all translatable elements of the page or only those elements which are new or have been updated.
 - e. Click **Export XLIFF**.
5. Follow the on-screen instructions.



See Also:

["Seeding Translatable Text"](#)

About Including XLIFF Target Elements

When Oracle Application Express generates an XLIFF document, each document contains multiple translation units. Each translation unit consists of a source element and a target element. The XLIFF document can be generated with both the source and target elements for each translation unit. You have the option of generating a file containing only source elements. The updated translations will be applied from the target elements of the translation units.

Step 3: Translate the XLIFF File

After you export a translatable file to XLIFF format, you can translate it into the appropriate languages.

Because XLIFF is an open standard XML file for exchanging translations, most translation vendors should support it. Oracle Application Express only supports XLIFF files encoded in UTF-8 character sets. In other words, it exports XLIFF files for translation in UTF-8 and assumes that the translated XLIFF files are in the same character set.

Translation is a time-consuming task. Oracle Application Express supports incremental translation so that application development can be done in parallel with the translation. An XLIFF file can be translated and uploaded to Oracle Application Express even when only part of the XLIFF file is translated. For strings that have no translation in the corresponding translated application, Oracle Application Express uses the corresponding ones in the primary language.

 **See Also:**

For more information about the XLIFF and the XLIFF 1.0 specification, see:

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

Step 4: Upload and Apply a Translated XLIFF Document

After your XLIFF document has been translated, the next step is to upload and then apply it.

- [Uploading a Translated XLIFF Document](#)
- [Applying an Uploaded XLIFF Document](#)
- [Deleting an Uploaded XLIFF Document](#)

Uploading a Translated XLIFF Document

To upload a translated XLIFF document:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.

The Translate page appears.

2. Click **Apply XLIFF translation files**.

The XLIFF Translation Files page appears.

3. Click **Upload Files**.
4. On the XLIFF Upload page:
 - a. Locate the file to be uploaded.
 - b. Click **Upload**.

The uploaded document appears on the XLIFF Translation Files page.

Applying an Uploaded XLIFF Document

After you upload an XLIFF document, the next step is to apply the XLIFF document and then publish the translated application. When you apply an XLIFF document, the Application Express engine parses the file and then updates the translation tables with the new translatable text.

Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternate languages.

Remember that to run an application in an alternative language, you must run it with globalization settings that cause an alternative language version to display. For

example, if the language is derived from the browser language, you must set the browser language to the same language as the translated application.

To apply a translated XLIFF document:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.

The Translate Application page appears.

2. Click **Apply XLIFF translation files**.

The XLIFF Translation Files page appears.

3. To apply the XLIFF files:
 - a. Select the XLIFF files you want to apply.
 - b. From Apply to Translation, choose a translation mapping.
 - c. Click **Apply Checked**.

The XLIFF file is parsed and the contents is applied to the translation repository. Once this process is complete, you must publish the application for each translation.



See Also:

- ["Specifying the Primary Language for an Application"](#)
- ["Step 5: Publish the Application"](#)

Deleting an Uploaded XLIFF Document

To delete an uploaded XLIFF document:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.

The Translate page appears.

2. Click **Apply XLIFF translation files**.

The XLIFF Translation Files page appears.

3. In the XLIFF Files repository, select the check box to the left of the Filename column.
4. Click **Delete Checked**.



See Also:

["Specifying the Primary Language for an Application"](#)

Step 5: Publish the Application

Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternate languages.

Remember that to run an application in an alternative language, you must run it with globalization settings that cause an alternative language version to display. For example, if the language is derived from the browser language, you must set the browser language to the same language as the translated application.

To publish an application:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.

The Translate Application page appears.

2. Click **Publish translated applications**.

The XLIFF Translation Files page appears

3. Select the items you want to publish and click **Publish**.

You should verify the existence of the translated application after it is published. Translated applications do not display in the Available Applications list on the App Builder home page. Instead, use the Application Navigate list on the left side of the page.

Note that in order for a translated application to appear in App Builder, you must ensure that you have correctly configured the application Globalization attributes.



See Also:

["Specifying the Primary Language for an Application"](#)

Manually Editing a Translation

Once you have mapped the target language and seeded the translatable text, you manually edit a translation.

To manually edit a translation:

1. Define the application language.

2. Seed the translatable text.
3. Navigate to the Translatable Text:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.
4. Under Translation Utilities, click **Translation Repository**.
The Translatable Text page appears.
5. Select a Language Mapping or page and click **Set**.
6. To further customize the report, use the Search bar at the top of the page.
7. To edit translatable text:
 - a. Click the **Edit** icon.
The Translatable Text page appears.
 - b. Translate From Text - Identifies the primary language text.
 - c. Translate to Text - Identifies the translated text.
 - d. Update all occurrences of this string - Enable this option to update all occurrences of the string in the Translation Repository.
 - e. Click **Apply Changes**.
8. To access a Grid Edit page:
 - a. Click **Grid Edit**.
The Translatable Text - Grid Edit page appears.
 - b. Translate To - Enter the translated text..
 - c. Click **Apply Changes**.



See Also:

- ["Step 1: Define Application Language"](#)
- ["Seeding Translatable Text"](#)
- ["Customizing Interactive Reports in a Running Application"](#)

Translating Messages

Learn when and how to translate messages in an Oracle Application Express application.

- [About Translating Messages](#)
You translate messages when your application includes PL/SQL regions, processes, package, procedure or function.
- [Translating Messages Used in PL/SQL Procedures](#)
Learn about how to translate messages used in PL/SQL procedures.

- [Interactive Grid Messages Requiring Translation](#)
Interactive Grid Messages Requiring Translation lists the interactive grid messages that require translation.
- [Translating Messages Used Internally by Oracle Application Express](#)
If your application uses a language that is not among the ten languages into which Oracle Application Express is translated, you must translate messages displayed by the Application Express reporting engine.

About Translating Messages

You translate messages when your application includes PL/SQL regions, processes, package, procedure or function.

You may need to translate messages if your application:

- Includes PL/SQL regions or PL/SQL processes, or calls a PL/SQL package, procedure, or function. If it does, you may need to translate the generated HTML.
- Uses a language that is not one of the ten languages into which Oracle Application Express is translated. If it does, you may need to translate messages used in reports.

Translating Messages Used in PL/SQL Procedures

Learn about how to translate messages used in PL/SQL procedures.

- [About Translating Messages Used in PL/SQL Procedures](#)
- [Defining a Translation Message](#)

See Also:

`APEX_LANG.MESSAGE` in *Oracle Application Express API Reference*

About Translating Messages Used in PL/SQL Procedures

If your application includes PL/SQL regions or PL/SQL processes or calls PL/SQL package, procedures, or functions, you may need to translate generated HTML. First, you define each message on the Translatable Messages page. Second, you use the `APEX_LANG.MESSAGE` API to translate the messages from PL/SQL stored procedures, functions, triggers, or packaged procedures and functions.

Defining a Translation Message

You create translatable messages on the Translate Messages page.

To define a new translation message:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

- c. Click **Shared Components**.
 - d. Under Globalization, click **Text Messages**.
2. On the Translate Messages page, click **Create Text Message**.
3. On Identify Text Message:
 - a. **Name** - Enter a name to identify the message.
 - b. **Language** - Select the language for which the message would be used.
 - c. **Used in JavaScript** - Select **Yes** if the message is used in APEX.LANG API calls in JavaScript code.
 - d. **Text** - Enter the text to be returned when the text message is called.
For example, you could define the message GREETING_MSG in English as:

```
Good morning %0
```

 Or, you could define the message GREETING_MSG in German as:

```
Guten Tag %0
```
4. Click **Create Text Message**.

Interactive Grid Messages Requiring Translation

Interactive Grid Messages Requiring Translation lists the interactive grid messages that require translation.

Table 21-2 Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|-------------------------|------------------------|
| APEX.GV.BREAK_COLLAPSE | Collapse control break |
| APEX.GV.BREAK_EXPAND | Expand control break |
| APEX.GV.DELETED_COUNT | %0 rows deleted |
| APEX.GV.DUP_REC_ID | Duplicate identity |
| APEX.GV.FIRST_PAGE | First |
| APEX.GV.LAST_PAGE | Last |
| APEX.GV.LOAD_MORE | Load More Rows |
| APEX.GV.NEXT_PAGE | Next |
| APEX.GV.PAGE_RANGE_XY | %0 - %1 |
| APEX.GV.PAGE_RANGE_XYZ | %0 - %1 of %2 |
| APEX.GV.PREV_PAGE | Previous |
| APEX.GV.ROW_ADDED | Added |
| APEX.GV.ROW_CHANGED | Changed |
| APEX.GV.ROW_DELETED | Deleted |
| APEX.GV.SELECT_PAGE_N | Page %0 |
| APEX.GV.SELECTION_COUNT | %0 rows selected |
| APEX.GV.SORT_ASCENDING | Sort Ascending |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|---------------------------------------|--------------------------------|
| APEX.GV.SORT_ASCENDING_ORDER | Sort Ascending %0 |
| APEX.GV.SORT_DESCENDING | Sort Descending |
| APEX.GV.SORT_DESCENDING_ORDER | Sort Descending %0 |
| APEX.GV.SORT_OFF | Don't Sort |
| APEX.GV.TOTAL_PAGES | Total %0 |
| APEX.IG.ACC_LABEL | Interactive Grid %0 |
| APEX.IG.ACTIONS | Actions |
| APEX.IG.ADD | Add |
| APEX.IG.ADD_ROW | Add Row |
| APEX.IG.AGGREGATE | Aggregate |
| APEX.IG.AGGREGATION | Aggregation |
| APEX.IG.ALL | All |
| APEX.IG.ALL_TEXT_COLUMNS | All Text Columns |
| APEX.IG.ALTERNATIVE | Alternative |
| APEX.IG.AND | and |
| APEX.IG.APPROX_COUNT_DISTINCT | Approx. Count Distinct |
| APEX.IG.APPROX_COUNT_DISTINCT_OVERALL | Overall Approx. Count Distinct |
| APEX.IG.AREA | Area |
| APEX.IG.ASCENDING | Ascending |
| APEX.IG.AUTHORIZATION | Authorization |
| APEX.IG.AUTO | Auto |
| APEX.IG.AVG | Average |
| APEX.IG.AVG_OVERALL | Overall Average |
| APEX.IG.AXIS_LABEL_TITLE | Label Axis Title |
| APEX.IG.AXIS_VALUE_DECIMAL | Decimal Places |
| APEX.IG.AXIS_VALUE_TITLE | Value Axis Title |
| APEX.IG.BACKGROUND_COLOR | Background Color |
| APEX.IG.BAR | Bar |
| APEX.IG.BETWEEN | between |
| APEX.IG.BOTH | Both |
| APEX.IG.BUBBLE | Bubble |
| APEX.IG.CANCEL | Cancel |
| APEX.IG.CASE_SENSITIVE | Case Sensitive |
| APEX.IG.CASE_SENSITIVE_WITH_BRACKETS | (Case Sensitive) |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--------------------------------|--|
| APEX.IG.CHANGE_VIEW | Change View |
| APEX.IG.CHANGES_SAVED | Changes saved |
| APEX.IG.CHART | Chart |
| APEX.IG.CHART_VIEW | Chart View |
| APEX.IG.CLOSE_COLUMN | Close |
| APEX.IG.COLOR_BLUE | Blue |
| APEX.IG.COLOR_GREEN | Green |
| APEX.IG.COLOR_ORANGE | Orange |
| APEX.IG.COLOR_RED | Red |
| APEX.IG.COLOR_YELLOW | Yellow |
| APEX.IG.COLORS | Colors |
| APEX.IG.COLUMN | Column |
| APEX.IG.COLUMN_CONTEXT | Column %0 |
| APEX.IG.COLUMN_TYPE | Column Purpose |
| APEX.IG.COLUMNS | Columns |
| APEX.IG.COMPLEX | Complex |
| APEX.IG.COMPUTE | Compute |
| APEX.IG.CONTAINS | contains |
| APEX.IG.CONTROL_BREAK | Control Break |
| APEX.IG.COUNT | Count |
| APEX.IG.COUNT_DISTINCT | Count Distinct |
| APEX.IG.COUNT_DISTINCT_OVERALL | Overall Count Distinct |
| APEX.IG.COUNT_OVERALL | Overall Count |
| APEX.IG.CREATE_X | Create %0 |
| APEX.IG.DATA | Data |
| APEX.IG.DATA_TYPE | Data Type |
| APEX.IG.DATE | Date |
| APEX.IG.DAYS | days |
| APEX.IG.DEFAULT_SETTINGS | Default Settings |
| APEX.IG.DEFAULT_TYPE | Default Type |
| APEX.IG.DELETE | Delete |
| APEX.IG.DELETE_REPORT_CONFIRM | Are you sure you would like to delete this report? |
| APEX.IG.DELETE_ROW | Delete Row |
| APEX.IG.DELETE_ROWS | Delete Rows |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--------------------------------|-------------------------|
| APEX.IG.DESCENDING | Descending |
| APEX.IG.DETAIL | Detail |
| APEX.IG.DETAIL_VIEW | Detail View |
| APEX.IG.DIRECTION | Direction |
| APEX.IG.DISABLED | Disabled |
| APEX.IG.DOES_NOT_CONTAIN | does not contain |
| APEX.IG.DOES_NOT_START_WITH | does not start with |
| APEX.IG.DONUT | Donut |
| APEX.IG.DOWNLOAD | Download |
| APEX.IG.DOWNLOAD_FORMAT | Choose Format |
| APEX.IG.DUPLICATE_AGGREGATION | Duplicate Aggregation |
| APEX.IG.DUPLICATE_CONTROLBREAK | Duplicate Control Break |
| APEX.IG.DUPLICATE_ROW | Duplicate Row |
| APEX.IG.DUPLICATE_ROWS | Duplicate Rows |
| APEX.IG.EDIT | Edit |
| APEX.IG.EDIT_CHART | Edit Chart |
| APEX.IG.EDIT_GROUP_BY | Edit Group By |
| APEX.IG.EMAIL_BCC | Blind Copy (bcc) |
| APEX.IG.EMAIL_BODY | Message |
| APEX.IG.EMAIL_CC | Copy (cc) |
| APEX.IG.EMAIL_SENT | Email sent. |
| APEX.IG.EMAIL_SUBJECT | Subject |
| APEX.IG.EMAIL_TO | Recipient (to) |
| APEX.IG.ENABLED | Enabled |
| APEX.IG.EQUALS | equals |
| APEX.IG.EXAMPLE | Example |
| APEX.IG.EXPRESSION | Expression |
| APEX.IG.FD_TYPE | Type |
| APEX.IG.FILTER | Filter |
| APEX.IG.FILTER_WITH_DOTS | Filter... |
| APEX.IG.FILTERS | Filters |
| APEX.IG.FIRST | First |
| APEX.IG.FLASHBACK | Flashback |
| APEX.IG.FORMAT | Format |
| APEX.IG.FORMAT_CSV | CSV |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--------------------------------------|---|
| APEX.IG.FORMAT_HTML | HTML |
| APEX.IG.FORMATMASK | Format Mask |
| APEX.IG.FREEZE | Freeze |
| APEX.IG.FUNCTIONS_AND_OPERATORS | Functions and Operators |
| APEX.IG.FUNNEL | Funnel |
| APEX.IG.GO | Go |
| APEX.IG.GREATER_THAN | greater than |
| APEX.IG.GREATER_THAN_OR_EQUALS | greater than or equals |
| APEX.IG.GRID | Grid |
| APEX.IG.GRID_VIEW | Grid View |
| APEX.IG.GROUP | Group |
| APEX.IG.GROUP_BY | Group By |
| APEX.IG.GROUP_BY_VIEW | Group By View |
| APEX.IG.HD_TYPE | Condition Type |
| APEX.IG.HEADING | Heading |
| APEX.IG.HEADING_ALIGN | Heading Alignment |
| APEX.IG.HELP | Help |
| APEX.IG.HELP.ACTIONS.EDITING | <p><p>You can insert, update, and delete data directly within this interactive grid.</p></p> <p><p>Insert a new row by clicking the Add Row button.</p></p> <p><p>Edit existing data by double-clicking a specific cell. For larger editing work, click Edit to enter editing mode. In editing mode, you can single-click or use the keyboard to edit specific cells.</p></p> <p><p>Use the Change menu to duplicate and delete rows. To enable the Change menu, use the check boxes to select one or more rows.</p></p> <p><p>Duplicate a selected row by clicking the Change menu and selecting Duplicate Rows. Delete a selected row by clicking the Change menu and selecting Delete Row.</p></p> |
| APEX.IG.HELP.ACTIONS.EDITING_HEADING | Editing Capabilities |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--|---|
| APEX.IG.HELP.ACTIONS.INTRO | <p><p>An interactive grid presents a set of data in a searchable, customizable report. You can perform numerous operations to limit the records returned, and change the way the data is displayed.</p></p> <p><p>Use the Search field to filter the records returned. Click Actions to access numerous options for modifying the report layout, or use the Column Heading menus on displayed columns.</p></p> <p><p>Use Report Settings to save your customizations to a report. You can also download the data from the report to an external file or email the data to yourself or others.</p></p> <p><p>To learn more, see Using Interactive Grids" in Oracle Application Express End User's Guide</p></p> |
| APEX.IG.HELP.ACTIONS.INTRO_HEADING | Overview |
| APEX.IG.HELP.ACTIONS.REPORTING | <p><p>You can customize the interactive grid to display data in various different ways using the built-in capabilities.</p></p> <p><p>Use the Column Heading menus or the Actions menu to determine which columns to display, in what sequence, and freeze columns. You can also define various data filters and sort the data returned.</p></p> <p><p>Use the View button (adjacent to the Search field) to access other data views that may have been defined by the application developer. You can also create a chart or view an existing chart.</p></p> <p><p>Note: Click Help in the interactive grid dialogs to obtain more detailed information on the selected function.</p></p> |
| APEX.IG.HELP.ACTIONS.REPORTING_HEADING | Reporting Capabilities |
| APEX.IG.HELP.ACTIONS_TITLE | Interactive Grid Help |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|------------------------------|--|
| APEX.IG.HELP.AGGREGATE | <p><p>Use this dialog to aggregate columns. Aggregated values display at the bottom of the data, or if Control Breaks are defined, at the bottom of each break.</p></p> <p><p>Aggregation List The Aggregation list displays defined aggregations. Disable an existing aggregation by deselecting it. Click Add (&plus;) to create a new aggregation, or Delete (&minus;) to remove an existing aggregation.</p></p> <p><p>Aggregation Settings Use the form on the right to define the aggregation. Select the Column name and Aggregation type. Optionally, enter a tooltip for the aggregation. If you have defined a Control Break, selecting Show Overall Value displays the overall average, total, or similar value at the bottom of the data.</p></p> <p><p>Note: Access the Aggregation dialog in the Actions menu or by clicking the column heading and sum(&sum;).</p></p> |
| APEX.IG.HELP.AGGREGATE_TITLE | Aggregation Help |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--------------------|--|
| APEX.IG.HELP.CHART | <p><p>Use this dialog to define a chart which displays as a separate data view. Select a chart Type, configure the chart settings, and click Save.</p></p> <p><p>Chart Settings</p></p> <p>The chart attributes that display vary depending on the chart type. A number of attributes can be entered to define the chart. Attributes marked with a red asterisk are mandatory.</p></p> <p><p>Below are all available attributes across the different chart types (in alphabetical order):</p></p> <ul style="list-style-type: none"> Aggregation - Select how to aggregate the associated chart values. Close - Select the column that contains the daily stock close price (Stock chart only). Decimal Places - Enter the number of decimal places to which the values are rounded. Direction - In relation to the Sort By attribute, specify whether the data is sorted in ascending or descending values. High - Select the column that contains the high value (Range and Stock charts only). Label - Select the column that contains the text for each data point. Label Axis Title - Enter the title that displays on the label axis. Low - Select the column that contains the low value (Range and Stock charts only). Nulls - In relation to the Sort By attribute, specify how you want records with null values to be sorted in relation to records with non null values. Open - Select the column that contains the daily stock opening price (Stock chart only). Orientation - Select whether the chart elements, such as bars, display vertically or horizontally. Series - Select the column used for defining your multi-series dynamic query. Stack - Specify whether the data items are stacked. Sort By - Select whether the chart is sorted by the label or the value(s). Target - Select the column to be used for defining the target value on this chart. When set, the Value attribute defines the filled area |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|-----------------------------|--|
| | <p>within the slice and the Target represents the value of the whole slice (Funnel chart only).</p> <p>Value - Select the column that contains the data to be plotted.</p> <p>Value Axis Title - Enter the title that displays on the value axis.</p> <p>Volume - Select the column that contains the daily stock volume (Stock chart only).</p> <p>X - Select the column that contains the x-axis value for this chart (Bubble and Scatter charts only).</p> <p>Y - Select the column that contains the y-axis value for this chart (Bubble and Scatter charts only).</p> <p>Z - Select the column that contains the bar width or bubble radius (Bar, Bubble, and Range charts only)</p> |
| APEX.IG.HELP.CHART_TITLE | Chart Help |
| APEX.IG.HELP.COLUMNNS | <p><p> Use this dialog to choose which columns display and in what order.</p></p> <p><p>Hide a column by deselecting it. </p> <p>Reorder columns by clicking Move Up (&uarr;) or Move Down (&darr;). </p> <p>Use the drop down selector to list All columns, Displayed columns, or Not Displayed columns.</p></p> <p><p>Optionally, use the form to specify the minimum width of a column in pixels.</p></p> <p><p> Note: You can also reorder displayed columns by clicking the drag handle (at the start of the column heading) and dragging the column left or right. You can also change the column width of displayed columns by selecting the column separator, between headings, and moving it left or right.</p></p> |
| APEX.IG.HELP.COLUMNNS_TITLE | Columns Help |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|----------------------------------|--|
| APEX.IG.HELP.COMPUTE | <p><p> Use this dialog to define additional columns based on mathematical and functional computations performed against existing columns.</p></p> <p><p>Computation List The Computation list displays defined computations. Disable an existing computation by deselecting it. Click Add (&plus) to add a new computation, or Delete (&minus;) to remove an existing computation.</p></p> <p><p>Computation Settings Use the form to define the computation. Enter the column details such as heading, label, and select alignment settings. Use the Expression textarea to enter the column(s) and associated functions for the computation. Select the appropriate data type, and optionally a format mask, for the new column.</p></p> |
| APEX.IG.HELP.COMPUTE_TITLE | Compute Help |
| APEX.IG.HELP.CONTROL_BREAK | <p><p>Use this dialog to define a control break on one or more columns.</p></p> <p><p>Control Break List The Control Break list displays defined control breaks. Disable an existing control break column by deselecting it. Click Add (&plus) to include a new column in the control break, or Delete (&minus;) to remove an existing column from the control break. To reorder columns, click Move Up (&uarr;) or Move Down (&darr;) to move the selected column up and down relative to other columns.</p></p> <p><p>Control Break Settings Use the form to define the control break column. Selected a control break column, the sort direction, and how to order null columns (columns with no value).</p></p> <p><p> Note: When viewing the interactive grid, you can define a control break by clicking a Column Heading and selecting the control break icon.</p></p> |
| APEX.IG.HELP.CONTROL_BREAK_TITLE | Control Break Help |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|------------------------------|---|
| APEX.IG.HELP.DOWNLOAD | <p><p>Use this dialog to download all the current rows to an external file. The file will contain only the currently displayed columns, using any filters and sorts applied to the data.</p></p> <p><p>Select the file format and click Download. </p> <p>Note: CSV will not include text formatting such as aggregates and control breaks.</p></p> <p><p>To email the file, select Send as Email and enter the email details (Recipient, Subject and Message).</p></p> |
| APEX.IG.HELP.DOWNLOAD_TITLE | Download Help |
| APEX.IG.HELP.FILTER | <p><p>Use this dialog to configure data filters which limit the rows returned.</p></p> <p><p>Filter List </p> <p>The Filter list displays defined filters. Disable an existing filter by deselecting it. </p> <p>Click Add (&plus;) to create a new filter, or Delete (&minus;) to remove an existing filter.</p></p> <p><p>Filter Settings </p> <p>Use the form to define the filter properties. </p> <p>Select the appropriate filter type: </p> <p>&nbsp; ; &nbsp; ; &nbsp; ; Row - filter for a term in any filterable column. </p> <p>&nbsp; ; &nbsp; ; &nbsp; ; Column - filter a specific column with a specified operator and value.</p></p> <p><p> Note: When viewing the interactive grid, you can define row filters by typing directly into the Search field. Click Select Columns to Search to limit the search to a specific column. Alternately, open a Column Heading menu and select a value to create a column filter.</p></p> |
| APEX.IG.HELP.FILTER_TITLE | Filter Help |
| APEX.IG.HELP.FLASHBACK | <p><p>Use this dialog to view the data as it existed at a previous point in time. </p></p> <p><p>Enter the number of minutes in the past to execute the flashback query.</p></p> |
| APEX.IG.HELP.FLASHBACK_TITLE | Flashback Help |
| APEX.IG.HELP.GROUP_BY_TITLE | Group By Help |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|------------------------------|---|
| APEX.IG.HELP.HIGHLIGHT | <p><p>Use this dialog to highlight rows or columns of data based on the condition entered.</p></p> <p><p>Highlight List The Highlight list displays defined highlights. Disable an existing highlight by deselecting it. </p> <p>Click Add (&plus;) to create a new highlight, or Delete (&minus;) to remove an existing highlight. </p> <p><p>Highlight Settings Use the form to define the highlight properties. </p> <p>Enter the name, select Row or Column, and select the HTML color codes for the background and text. </p> <p>Select the appropriate Condition Type to highlight specific data: </p> <p>&nbsp; ; &nbsp; ; &nbsp; ; Row - highlight the term in any column. </p> <p>&nbsp; ; &nbsp; ; &nbsp; ; Column - highlight within a specific column based on the specified operator and value.</p></p> |
| APEX.IG.HELP.HIGHLIGHT_TITLE | Highlight Help |
| APEX.IG.HELP.REPORT | <p><p>Use this dialog to save changes you have made to the current grid layout and configuration. </p> <p>Application developers can define multiple alternate report layouts. Where permissible, you and other end users can save a report as Public, which makes the report available to all other users of the grid. You can also save a report as a Private report that only you can view.</p></p> <p><p>Select from the available types and enter a name for the saved report.</p></p> |
| APEX.IG.HELP.REPORT_TITLE | Report Help |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|---------------------------------|--|
| APEX.IG.HELP.SORT | <p><p>Use this dialog to set the display order.</p></p> <p><p>Sort List</p></p> <p>The Sort dialog displays a list of configured sorting rules. </p> <p>Click Add (&plus;) to create a sort column, or Delete (&minus;) to remove a sort column. </p> <p>Click Move Up (&uarr;) and Move Down (&darr;) to move the selected sort column up and down relative to the other sort columns.</p></p> <p><p>Sort Settings</p></p> <p>Select a sort column, the sort direction, and how to order null columns (columns with no value).</p></p> <p><p> Note: Data can be sorted by columns which are not displayed; however, not all columns may be sortable.</p></p> <p> Displayed columns can be sorted by pressing the up (ascending) or down (descending) arrows at the end of the column headings. To add a subsequent column to an existing sort, hold the Shift key and click the up or down arrow.</p></p> |
| APEX.IG.HELP.SORT_TITLE | Sort Help |
| APEX.IG.HELP.SUBSCRIPTION_TITLE | Subscription Help |
| APEX.IG.HIDE | Hide |
| APEX.IG.HIGH_COLUMN | High |
| APEX.IG.HIGHLIGHT | Highlight |
| APEX.IG.HORIZONTAL | Horizontal |
| APEX.IG.HOURS | hours |
| APEX.IG.ICON | Icon |
| APEX.IG.ICON_VIEW | Icon View |
| APEX.IG.IN | in |
| APEX.IG.IN_THE_LAST | in the last |
| APEX.IG.IN_THE_NEXT | in the next |
| APEX.IG.INACTIVE_SETTING | Inactive Setting |
| APEX.IG.INACTIVE_SETTINGS | Inactive Settings |
| APEX.IG.INTERNAL_ERROR | An internal error has occurred while processing the Interactive Grid request. |
| APEX.IG.INVALID_DATE_FORMAT | Invalid Date Format |
| APEX.IG.INVALID_SETTING | Invalid Setting |
| APEX.IG.INVALID_SETTINGS | Invalid Settings |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|------------------------------------|--|
| APEX.IG.INVALID_SORT_BY | Sort By has been set to %0, but no column as been selected for %0. |
| APEX.IG.INVALID_VALUE | Invalid Value |
| APEX.IG.INVISIBLE | Not Displayed |
| APEX.IG.IS_NOT_NULL | is not empty |
| APEX.IG.IS_NULL | is empty |
| APEX.IG.LABEL | Label |
| APEX.IG.LABEL_COLUMN | Label |
| APEX.IG.LAST | Last |
| APEX.IG.LAST.DAY | Last Day |
| APEX.IG.LAST.HOUR | Last Hour |
| APEX.IG.LAST.MINUTE | Last Minute |
| APEX.IG.LAST.MONTH | Last Month |
| APEX.IG.LAST.WEEK | Last Week |
| APEX.IG.LAST.X_DAYS | Last %0 Days |
| APEX.IG.LAST.X_HOURS | Last %0 Hours |
| APEX.IG.LAST.X_MINUTES | Last %0 Minutes |
| APEX.IG.LAST.X_MONTHS | Last %0 Months |
| APEX.IG.LAST.X_WEEKS | Last %0 Weeks |
| APEX.IG.LAST.X_YEARS | Last %0 Years |
| APEX.IG.LAST.YEAR | Last Year |
| APEX.IG.LAYOUT_ALIGN | Cell Alignment |
| APEX.IG.LAYOUT_USEGROUPFOR | Use Group For |
| APEX.IG.LESS_THAN | less than |
| APEX.IG.LESS_THAN_OR_EQUALS | less than or equals |
| APEX.IG.LINE | Line |
| APEX.IG.LINE_WITH_AREA | Line with Area |
| APEX.IG.LISTAGG | Listagg |
| APEX.IG.LOW_COLUMN | Low |
| APEX.IG.MAILADDRESSES_COMMASEP | Separate multiple Addresses with commas |
| APEX.IG.MATCHES_REGULAR_EXPRESSION | matches regular expression |
| APEX.IG.MAX | Maximum |
| APEX.IG.MAX_OVERALL | Overall Maximum |
| APEX.IG.MEDIAN | Median |
| APEX.IG.MEDIAN_OVERALL | Overall Median |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--------------------------|--|
| APEX.IG.MIN | Minimum |
| APEX.IG.MIN_OVERALL | Overall Minimum |
| APEX.IG.MINUTES | minutes |
| APEX.IG.MINUTES_AGO | Minutes ago |
| APEX.IG.MONTHS | months |
| APEX.IG.MORE_DATA_FOUND | The data contains more than %0 rows which exceeds the maximum allowed. Please apply additional filters in order to view the results. |
| APEX.IG.NAME | Name |
| APEX.IG.NAMED_REPORT | Named Report |
| APEX.IG.NEXT.DAY | Next Day |
| APEX.IG.NEXT.HOUR | Next Hour |
| APEX.IG.NEXT.MINUTE | Next Minute |
| APEX.IG.NEXT.MONTH | Next Month |
| APEX.IG.NEXT.WEEK | Next Week |
| APEX.IG.NEXT.X_DAYS | Next %0 Days |
| APEX.IG.NEXT.X_HOURS | Next %0 Hours |
| APEX.IG.NEXT.X_MINUTES | Next %0 Minutes |
| APEX.IG.NEXT.X_MONTHS | Next %0 Months |
| APEX.IG.NEXT.X_WEEKS | Next %0 Weeks |
| APEX.IG.NEXT.X_YEARS | Next %0 Years |
| APEX.IG.NEXT.YEAR | Next Year |
| APEX.IG.NO_DATA_FOUND | No data found |
| APEX.IG.NOT_BETWEEN | not between |
| APEX.IG.NOT_EQUALS | not equals |
| APEX.IG.NOT_EXIST | Region with ID %0 is not an Interactive Grid region or does not exist in application %1. |
| APEX.IG.NOT_IN | not in |
| APEX.IG.NOT_IN_THE_LAST | not in the last |
| APEX.IG.NOT_IN_THE_NEXT | not in the next |
| APEX.IG.NULLS | Nulls |
| APEX.IG.NUMBER | Numeric |
| APEX.IG.OFF | Off |
| APEX.IG.ON | On |
| APEX.IG.ONE_MINUTE_AGO | 1 minute ago |
| APEX.IG.OPEN_COLORPICKER | Open Color Picker: %0 |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--|--|
| APEX.IG.OPEN_COLUMN | Open |
| APEX.IG.OPERATOR | Operator |
| APEX.IG.ORIENTATION | Orientation |
| APEX.IG.PIE | Pie |
| APEX.IG.PIVOT | Pivot |
| APEX.IG.PIVOT_VIEW | Pivot View |
| APEX.IG.PLACEHOLDER_INVALUES | Separate values with "%0" |
| APEX.IG.POLAR | Polar |
| APEX.IG.POSITION_CENTER | Center |
| APEX.IG.POSITION_END | End |
| APEX.IG.POSITION_START | Start |
| APEX.IG.PRIMARY | Primary |
| APEX.IG.PRIMARY_DEFAULT | Primary Default |
| APEX.IG.PRIMARY_REPORT | Primary Report |
| APEX.IG.RADAR | Radar |
| APEX.IG.RANGE | Range |
| APEX.IG.REFRESH | Refresh |
| APEX.IG.REFRESH_ROW | Refresh Row |
| APEX.IG.REFRESH_ROWS | Refresh Rows |
| APEX.IG.REMOVE_CONTROL | Remove %0 |
| APEX.IG.REPORT | Report |
| APEX.IG.REPORT.DELETED | Report deleted |
| APEX.IG.REPORT.SAVED.ALTERNATIVE | Alternative report saved for all users |
| APEX.IG.REPORT.SAVED.DEFAULT | Default report saved for all users |
| APEX.IG.REPORT.SAVED.PRIVATE | Private report saved |
| APEX.IG.REPORT.SAVED.PUBLIC | Public report saved for all users |
| APEX.IG.REPORT_DATA_AS_OF.X.MINUTES_AGO | Report data as of %0 minutes ago |
| APEX.IG.REPORT_DATA_AS_OF.ONE_MINUTE_AGO | Report data as of 1 minute ago |
| APEX.IG.REPORT_EDIT | Report - Edit |
| APEX.IG.REPORT_SAVE_AS | Report - Save As |
| APEX.IG.REPORT_SETTINGS | Report Settings |
| APEX.IG.REPORT_STATIC_ID_DOES_NOT_EXIST | Saved Interactive Grid with static ID %0 does not exist. |
| APEX.IG.REPORT_VIEW | Report View |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|----------------------------------|--|
| APEX.IG.RESET | Reset |
| APEX.IG.REVERT_CHANGES | Revert Changes |
| APEX.IG.REVERT_ROWS | Revert Rows |
| APEX.IG.ROW | Row |
| APEX.IG.ROW_ACTIONS | Row Actions |
| APEX.IG.ROW_ACTIONS_FOR | Actions for row %0 |
| APEX.IG.ROW_COLUMN_CONTEXT | Row %0 Column %1 |
| APEX.IG.ROW_CONTEXT | Row %0 |
| APEX.IG.ROWS_PER_PAGE | Rows Per Page |
| APEX.IG.SAVE | Save |
| APEX.IG.SAVE_AS | Save As |
| APEX.IG.SAVE_REPORT_SETTINGS | Save Report Settings |
| APEX.IG.SAVED_REPORT_DEFAULT | Default |
| APEX.IG.SAVED_REPORT_PRIVATE | Private |
| APEX.IG.SAVED_REPORT_PUBLIC | Public |
| APEX.IG.SAVED_REPORTS | Saved Reports |
| APEX.IG.SCATTER | Scatter |
| APEX.IG.SEARCH | Search |
| APEX.IG.SEARCH.ALL_COLUMNS | Search: All Text Columns |
| APEX.IG.SEARCH.COLUMN | Search: %0 |
| APEX.IG.SEARCH.ORACLE_TEXT | Search: Full Text |
| APEX.IG.SEARCH_FOR.X | Search for '%0' |
| APEX.IG.SEL_ACTIONS | Selection Actions |
| APEX.IG.SELECT | - Select - |
| APEX.IG.SELECT_1_ROW_IN_MASTER | Select 1 row in the master region |
| APEX.IG.SELECT_COLUMNS_TO_SEARCH | Select columns to search |
| APEX.IG.SEND_AS_EMAIL | Send as Email |
| APEX.IG.SERIES_COLUMN | Series |
| APEX.IG.SHOW_OVERALL_VALUE | Show Overall Value |
| APEX.IG.SINGLE_ROW_VIEW | Single Row View |
| APEX.IG.SORT | Sort |
| APEX.IG.SORT_BY | Sort By |
| APEX.IG.SORT_ONLY_ONE_PER_COLUMN | You can define only one sort per column. |
| APEX.IG.SRV_CHANGE_MENU | Change Menu |
| APEX.IG.STACK | Stack |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--|---|
| APEX.IG.STARTS_WITH | starts with |
| APEX.IG.STOCK | Stock |
| APEX.IG.SUBSCRIPTION | Subscription |
| APEX.IG.SUM | Sum |
| APEX.IG.SUM_OVERALL | Overall Sum |
| APEX.IG.SUMMARY | Interactive Grid. Report: %0, View: %1. |
| APEX.IG.TARGET_COLUMN | Target |
| APEX.IG.TEXT_COLOR | Text Color |
| APEX.IG.TOGGLE | Toggle |
| APEX.IG.TOOLTIP | Tooltip |
| APEX.IG.TYPE | Type |
| APEX.IG.UNFREEZE | Unfreeze |
| APEX.IG.UNIT | Unit |
| APEX.IG.UNSAVED_CHANGES_CONTINUE_CONFIRM | There are unsaved changes. Do you want to continue? |
| APEX.IG.VALUE | Value |
| APEX.IG.VALUE_COLUMN | Value |
| APEX.IG.VALUE_REQUIRED | A value is required. |
| APEX.IG.VARCHAR2 | Text |
| APEX.IG.VERTICAL | Vertical |
| APEX.IG.VIEW | View |
| APEX.IG.VISIBLE | Displayed |
| APEX.IG.VOLUME_COLUMN | Volume |
| APEX.IG.WEEKS | weeks |
| APEX.IG.WIDTH | Minimum Column Width (Pixel) |
| APEX.IG.X.BETWEEN.Y.AND.Z | %0 between %1 and %2 |
| APEX.IG.X.CONTAINS.Y | %0 contains %1 |
| APEX.IG.X.DOES_NOT_CONTAIN.Y | %0 does not contain %1 |
| APEX.IG.X.EQUALS.Y | %0 equals %1 |
| APEX.IG.X.GREATER_THAN.Y | %0 greater than %1 |
| APEX.IG.X.GREATER_THAN_OR_EQUALS.Y | %0 greater than or equal to %1 |
| APEX.IG.X.IN.Y | %0 in %1 |
| APEX.IG.X.IN_THE_LAST.Y.DAYS | %0 in the last %1 days |
| APEX.IG.X.IN_THE_LAST.Y.HOURS | %0 in the last %1 hours |
| APEX.IG.X.IN_THE_LAST.Y.MINUTES | %0 in the last %1 minutes |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|--|----------------------------------|
| APEX.IG.X.IN_THE_LAST.Y.MONTHS | %0 in the last %1 months |
| APEX.IG.X.IN_THE_LAST.Y.WEEKS | %0 in the last %1 weeks |
| APEX.IG.X.IN_THE_LAST.Y.YEARS | %0 in the last %1 years |
| APEX.IG.X.IN_THE_LAST_DAY | %0 in the last day |
| APEX.IG.X.IN_THE_LAST_HOUR | %0 in the last hour |
| APEX.IG.X.IN_THE_LAST_MINUTE | %0 in the last minute |
| APEX.IG.X.IN_THE_LAST_MONTH | %0 in the last month |
| APEX.IG.X.IN_THE_LAST_WEEK | %0 in the last week |
| APEX.IG.X.IN_THE_LAST_YEAR | %0 in the last year |
| APEX.IG.X.IN_THE_NEXT.Y.DAYS | %0 in the next %1 days |
| APEX.IG.X.IN_THE_NEXT.Y.HOURS | %0 in the next %1 hours |
| APEX.IG.X.IN_THE_NEXT.Y.MINUTES | %0 in the next %1 minutes |
| APEX.IG.X.IN_THE_NEXT.Y.MONTHS | %0 in the next %1 months |
| APEX.IG.X.IN_THE_NEXT.Y.WEEKS | %0 in the next %1 weeks |
| APEX.IG.X.IN_THE_NEXT.Y.YEARS | %0 in the next %1 years |
| APEX.IG.X.IN_THE_NEXT_DAY | %0 in the next day |
| APEX.IG.X.IN_THE_NEXT_HOUR | %0 in the next hour |
| APEX.IG.X.IN_THE_NEXT_MINUTE | %0 in the next minute |
| APEX.IG.X.IN_THE_NEXT_MONTH | %0 in the next month |
| APEX.IG.X.IN_THE_NEXT_WEEK | %0 in the next week |
| APEX.IG.X.IN_THE_NEXT_YEAR | %0 in the next year |
| APEX.IG.X.IS_NOT_NULL | %0 is not empty |
| APEX.IG.X.IS_NULL | %0 is empty |
| APEX.IG.X.LESS_THAN.Y | %0 less than %1 |
| APEX.IG.X.LESS_THAN_OR_EQUALS.Y | %0 less than or equal to %1 |
| APEX.IG.X.LIKE.Y | %0 like %1 |
| APEX.IG.X.MATCHES_REGULAR_EXPRESSION.Y | %0 matches regular expression %1 |
| APEX.IG.X.MINUTES_AGO | %0 minutes ago |
| APEX.IG.X.NOT_BETWEEN.Y.AND.Z | %0 not between %1 and %2 |
| APEX.IG.X.NOT_EQUALS.Y | %0 not equals %1 |
| APEX.IG.X.NOT_IN.Y | %0 not in %1 |
| APEX.IG.X.NOT_IN_THE_LAST.Y.DAYS | %0 not in the last %1 days |
| APEX.IG.X.NOT_IN_THE_LAST.Y.HOURS | %0 not in the last %1 hours |
| APEX.IG.X.NOT_IN_THE_LAST.Y.MINUTES | %0 not in the last %1 minutes |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|-------------------------------------|-------------------------------|
| APEX.IG.X.NOT_IN_THE_LAST.Y.MONTHS | %0 not in the last %1 months |
| APEX.IG.X.NOT_IN_THE_LAST.Y.WEEKS | %0 not in the last %1 weeks |
| APEX.IG.X.NOT_IN_THE_LAST.Y.YEARS | %0 not in the last %1 years |
| APEX.IG.X.NOT_IN_THE_LAST_DAY | %0 not in the last day |
| APEX.IG.X.NOT_IN_THE_LAST_HOUR | %0 not in the last hour |
| APEX.IG.X.NOT_IN_THE_LAST_MINUTE | %0 not in the last minute |
| APEX.IG.X.NOT_IN_THE_LAST_MONTH | %0 not in the last month |
| APEX.IG.X.NOT_IN_THE_LAST_WEEK | %0 not in the last week |
| APEX.IG.X.NOT_IN_THE_LAST_YEAR | %0 not in the last year |
| APEX.IG.X.NOT_IN_THE_NEXT.Y.DAYS | %0 not in the next %1 days |
| APEX.IG.X.NOT_IN_THE_NEXT.Y.HOURS | %0 not in the next %1 hours |
| APEX.IG.X.NOT_IN_THE_NEXT.Y.MINUTES | %0 not in the next %1 minutes |
| APEX.IG.X.NOT_IN_THE_NEXT.Y.MONTHS | %0 not in the next %1 months |
| APEX.IG.X.NOT_IN_THE_NEXT.Y.WEEKS | %0 not in the next %1 weeks |
| APEX.IG.X.NOT_IN_THE_NEXT.Y.YEARS | %0 not in the next %1 years |
| APEX.IG.X.NOT_IN_THE_NEXT_DAY | %0 not in the next day |
| APEX.IG.X.NOT_IN_THE_NEXT_HOUR | %0 not in the next hour |
| APEX.IG.X.NOT_IN_THE_NEXT_MINUTE | %0 not in the next minute |
| APEX.IG.X.NOT_IN_THE_NEXT_MONTH | %0 not in the next month |
| APEX.IG.X.NOT_IN_THE_NEXT_WEEK | %0 not in the next week |
| APEX.IG.X.NOT_IN_THE_NEXT_YEAR | %0 not in the next year |
| APEX.IG.X.NOT_LIKE.Y | %0 not like %1 |
| APEX.IG.X.STARTS_WITH.Y | %0 starts with %1 |
| APEX.IG.X.COLUMN | X |
| APEX.IG.Y.COLUMN | Y |
| APEX.IG.YEARS | years |
| APEX.IG.Z.COLUMN | Z |
| APEX.IG_FORMAT_SAMPLE_1 | Monday, 12 January, 2016 |
| APEX.IG_FORMAT_SAMPLE_2 | January |
| APEX.IG_FORMAT_SAMPLE_3 | 16 hours ago |
| APEX.IG_FORMAT_SAMPLE_4 | in 16h |
| APEX.RV.NOT_GROUPED_LABEL | Other Columns |
| APEX.RV.REC_XY | Row %0 of %1 |
| APEX.RV.REC_X | Row %0 |
| APEX.RV.EXCLUDE_HIDDEN | Displayed Columns |

Table 21-2 (Cont.) Interactive Grid Messages Requiring Translation

| Message Name | English Text |
|-----------------------|---------------------|
| APEX.RV.EXCLUDE_NULL | Exclude Null Values |
| APEX.RV.PREV_RECORD | Previous |
| APEX.RV.NEXT_RECORD | Next |
| APEX.RV.SETTINGS_MENU | Settings Menu |
| APEX.RV.INSERT | Add |
| APEX.RV.DELETE | Delete |

Translating Messages Used Internally by Oracle Application Express

If your application uses a language that is not among the ten languages into which Oracle Application Express is translated, you must translate messages displayed by the Application Express reporting engine.

Oracle Application Express is translated into German, Spanish, French, Italian, Japanese, Korean, Brazilian Portuguese, Simplified Chinese, and Traditional Chinese.

For example, if you develop a Russian application and want to include report messages, such as pagination, in Russian, you must translate the strings used in messages displayed in reports.

- [Creating Text Messages](#)
- [Editing Text Messages](#)
- [Internal Messages Requiring Translation](#)
- [Interactive Report Messages Requiring Translation](#)

Creating Text Messages

To create a translation message:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Text Messages**.
2. On the Translate Messages page, click **Create Text Message**.
3. On Identify Text Message, specify the following:
 - a. **Name** - Enter the name of each message that must be translated.
 - b. **Language** - Select the language for which the message will be used.
 - c. **Text** - Enter the text to be returned when the text message is called. If the English text message contains positional substitution values (for example, %0, %1), ensure that your defined message also contains the same named and number of positional substitution values.

4. Click **Create Text Message**.

See Also:

- ["Internal Messages Requiring Translation"](#)
- ["Interactive Report Messages Requiring Translation"](#)

Editing Text Messages

To edit text messages:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Text Messages**.
2. On edit a text message:
 - a. Click the message name.
 - b. Make the appropriate edits.

If the English text message contains positional substitution values (for example, %0, %1), ensure that your defined message also contains the same named and number of positional substitution values.
 - c. Click **Apply Changes**.
3. To access a Grid Edit page:
 - a. Click **Grid Edit**.

The Grid Edit page appears.
 - b. Make the appropriate edits.

If the English text message contains positional substitution values (for example, %0, %1), ensure that your defined message also contains the same named and number of positional substitution values.
 - c. Click **Apply Changes**.

Tip:

To add a new message from the Grid Edit page, click **Add Text Message** at the bottom of the page.

Internal Messages Requiring Translation

Internal Messages Requiring Translation lists the internal messages that require translation.

Table 21-3 Internal Messages Requiring Translation

| Message Name | English Text |
|---|---|
| APEX.ACTIONS.TOGGLE | Toggle %0 |
| APEX.ACTIVE_STATE | (Active) |
| APEX.AUTHENTICATION.LOGIN_THROTTLE.COUNTER | Please wait %0 seconds to login again. |
| APEX.AUTHENTICATION.LOGIN_THROTTLE.ERROR | The login attempt has been blocked. |
| APEX.COMBOBOX.LIST_OF_VALUES | List of Values |
| APEX.COMBOBOX.SHOW_ALL_VALUES | Open list for: %0 |
| APEX.COMPLETED_STATE | (Completed) |
| APEX.CORRECT_ERRORS | Correct errors before saving. |
| APEX.DATA_HAS_CHANGED | Current version of data in database has changed since user initiated update process. current checksum = "%0" application checksum = "%1". |
| APEX.DATEPICKER.ICON_TEXT | Popup Calendar: %0 |
| APEX.DATEPICKER_VALUE_GREATER_MAX_DATE | #LABEL# is greater than specified maximum date %0. |
| APEX.DATEPICKER_VALUE_INVALID | #LABEL# does not match format %0. |
| APEX.DATEPICKER_VALUE_LESS_MIN_DATE | #LABEL# is less than specified minimum date %0. |
| APEX.DATEPICKER_VALUE_NOT_BETWEEN_MIN_MAX | #LABEL# is not between the valid range of %0 and %1. |
| APEX.DATEPICKER_VALUE_NOT_IN_YEAR_RANGE | #LABEL# is not within valid year range of %0 and %1. |
| APEX.DIALOG.CANCEL | Cancel |
| APEX.DIALOG.CLOSE | Close |
| APEX.DIALOG.HELP | Help |
| APEX.DIALOG.OK | OK |
| APEX.DIALOG.SAVE | Save |
| APEX.ERROR.PAGE_NOT_AVAILABLE | Sorry, this page isn't available |
| APEX.ERROR_MESSAGE_HEADING | Error Message |
| APEX.FILE_BROWSE.DOWNLOAD_LINK_TEXT | Download |
| APEX.GO_TO_ERROR | Go to error |
| APEX.ITEM.HELP_TEXT | Help Text: %0 |
| APEX.ITEM.NOT_FOUND | Item %0 not found. |
| APEX.ITEM_TYPE.SLIDER.VALUE_NOT_BETWEEN_MIN_MAX | #LABEL# is not between the valid range of %0 and %1. |

Table 21-3 (Cont.) Internal Messages Requiring Translation

| Message Name | English Text |
|--|--|
| APEX.ITEM_TYPE.SLIDER.VALUE_NOT_MULTIPLE_OF_STEP | #LABEL# is not a multiple of %0. |
| APEX.ITEM_TYPE.YES_NO.INVALID_VALUE | #LABEL# must match to the values %0 and %1. |
| APEX.ITEM_TYPE.YES_NO.NO_LABEL | No |
| APEX.ITEM_TYPE.YES_NO.YES_LABEL | Yes |
| APEX.MENU.CURRENT_MENU | current |
| APEX.MENU.OVERFLOW_LABEL | More... |
| APEX.MENU.PROCESSING | Loading |
| APEX.NUMBER_FIELD.VALUE_GREATER_MAX_VALUE | #LABEL# is greater than specified maximum %0. |
| APEX.NUMBER_FIELD.VALUE_INVALID | #LABEL# must be Numeric. |
| APEX.NUMBER_FIELD.VALUE_INVALID2 | #LABEL# does not match number format %0 (For example, %1). |
| APEX.NUMBER_FIELD.VALUE_LESS_MIN_VALUE | #LABEL# is less than specified minimum %0. |
| APEX.NUMBER_FIELD.VALUE_NOT_BETWEEN_MIN_MAX | #LABEL# is not between the valid range of %0 and %1. |
| APEX.PAGE.DUPLICATE_SUBMIT | This page was already submitted and can not be re-submitted. |
| APEX.PAGE_ITEM_IS_REQUIRED | #LABEL# must have some value. |
| APEX.POPUP_LOV.ICON_TEXT | Popup List of Values: %0. |
| APEX.PROCESSING | Processing |
| APEX.REGION.JQM_LIST_VIEW.LOAD_MORE | Load more |
| APEX.REGION.JQM_LIST_VIEW.SEARCH | Search |
| APEX.RICH_TEXT_EDITOR.ACCESSIBLE_LABEL | %0, rich text editor |
| APEX.SINCE.SHORT.DAYS_AGO | %0d |
| APEX.SINCE.SHORT.DAYS_FROM_NOW | in %0d |
| APEX.SINCE.SHORT.HOURS_AGO | %0h |
| APEX.SINCE.SHORT.HOURS_FROM_NOW | in %0h |
| APEX.SINCE.SHORT.MINUTES_AGO | %0m |
| APEX.SINCE.SHORT.MINUTES_FROM_NOW | in %0m |
| APEX.SINCE.SHORT.MONTHS_AGO | %0mo |
| APEX.SINCE.SHORT.MONTHS_FROM_NOW | in %0mo |
| APEX.SINCE.SHORT.SECONDS_AGO | %0s |
| APEX.SINCE.SHORT.SECONDS_FROM_NOW | in %0s |
| APEX.SINCE.SHORT.WEEKS_AGO | %0w |

Table 21-3 (Cont.) Internal Messages Requiring Translation

| Message Name | English Text |
|---|---|
| APEX.SINCE.SHORT.WEEKS_FROM_NOW | in %0w |
| APEX.SINCE.SHORT.YEARS_AGO | %0y |
| APEX.SINCE.SHORT.YEARS_FROM_NOW | in %0y |
| APEX.SUCCESS_MESSAGE_HEADING | Success Message |
| APEX.TABS.NEXT | Next |
| APEX.TABS.PREVIOUS | Previous |
| APEX.TB.TOOLBAR | Toolbar |
| APEX.TEMPLATE.EXPAND_COLLAPSE_NAV_LABEL | Expand / Collapse Navigation |
| APEX.TEMPLATE.EXPAND_COLLAPSE_SIDE_COLUMN_LABEL | Expand / Collapse Side Column |
| APEX.THEMES.INVALID_THEME_NUMBER | Theme number is invalid or theme is not a current UI theme. |
| APEX.UI.BACK_TO_TOP | Start of page |
| APEX.VALUE_REQUIRED | Value Required |
| APEX.WAIT_UNTIL_PAGE_LOADED | Please wait until the page is fully loaded and try again. |
| CHART_SERIES_ERROR | Chart series error %0 for %1. |
| FLOW.SINGLE_VALIDATION_ERROR | 1 error has occurred. |
| FLOW.VALIDATION_ERROR | %0 errors have occurred. |
| INVALID_CREDENTIALS | Invalid Login Credentials . |
| LAYOUT.CHART | Chart |
| LAYOUT.T_#EXPAND_COLLAPSE_NAV_LABEL# | Label for Expand / Collapse Navigation |
| LAYOUT.T_#EXPAND_COLLAPSE_SIDE_COLUMN_LABEL# | Label for Expand / Collapse Side Column |
| OUT_OF_RANGE | Invalid set of rows requested, the source data of the report has been modified. |
| PAGINATION.NEXT | Next |
| PAGINATION.NEXT_SET | Next Set |
| PAGINATION.PREVIOUS | Previous |
| PAGINATION.PREVIOUS_SET | Previous Set |
| REGION_NAME.NATIVE_JET_CHART | Chart |
| REPORT_TOTAL | report total |
| RESET | reset pagination |
| SHOW_ALL | Show All |
| SINCE_DAYS_AGO | %0 days ago |
| SINCE_DAYS_FROM_NOW | %0 days from now |

Table 21-3 (Cont.) Internal Messages Requiring Translation

| Message Name | English Text |
|--|---|
| SINCE_HOURS_AGO | %0 hours ago |
| SINCE_HOURS_FROM_NOW | %0 hours from now |
| SINCE_MINUTES_AGO | %0 minutes ago |
| SINCE_MINUTES_FROM_NOW | %0 minutes from now |
| SINCE_MONTHS_AGO | %0 months ago |
| SINCE_MONTHS_FROM_NOW | %0 months from now |
| SINCE_NOW | Now |
| SINCE_SECONDS_AGO | %0 seconds ago |
| SINCE_SECONDS_FROM_NOW | %0 seconds from now |
| SINCE_WEEKS_AGO | %0 weeks ago |
| SINCE_WEEKS_FROM_NOW | %0 weeks from now |
| SINCE_YEARS_AGO | %0 years ago |
| SINCE_YEARS_FROM_NOW | %0 years from now |
| TOTAL | Total |
| UPGRADE_CANDLESTICK_CHART | After upgrading, ensure the series attribute Label Column is mapped to a Date / Timestamp column. |
| WWV_FLOW_CREATE_MODEL_APP.CREATE_IG | Unable to create interactive grid page. %0 |
| WWV_FLOW_CUSTOMIZE.T_MESSAGE3 | You can personalize the appearance of this application by changing the Theme Style. Please select a Theme Style from the list below and click on Apply Changes. |
| WWV_FLOW_CUSTOMIZE.T_REGION_DISP | Region Display |
| WWV_FLOW_CUSTOMIZE.T_REMOVE_STYLE | Use Application Default Style |
| WWV_FLOW_CUSTOMIZE.T_THEME_STYLE | Appearance |
| WWV_FLOW_DATA_EXPORT.AGG_COLUMN_IDX_NOT_EXIST | The column index referenced in the aggregate %0 does not exist. |
| WWV_FLOW_DATA_EXPORT.COLUMN_BREAK_MUST_BE_IN_THE_BEGGINING | The column break needs to be in the beggining of the columns array. |
| WWV_FLOW_DATA_EXPORT.COLUMN_GROUP_ID_X_NOT_EXIST | The column group index referenced in %0 does not exist. |
| WWV_FLOW_DATA_EXPORT.HIGHLIGHT_COLUMN_IDX_NOT_EXIST | The column index referenced in the highlight %0 does not exist. |
| WWV_FLOW_DATA_EXPORT.PARENT_GROUP_ID_X_NOT_EXIST | The parent group index referenced in %0 does not exist. |
| WWV_FLOW_UTILITIES.CAL | Calendar |
| WWV_FLOW_UTILITIES.CLOSE | Close |
| WWV_FLOW_UTILITIES.OK | Ok |

Table 21-3 (Cont.) Internal Messages Requiring Translation

| Message Name | English Text |
|---|--|
| WWV_FLOW_WEB_SERVICES.AUTHENTICATION_FAILED | Authentication failed. |
| WWV_FLOW_WEB_SERVICES.NO_VALID_OAUTH_TOKEN | OAuth access token not available or expired. |
| WWV_FLOW_WEB_SERVICES.UNSUPPORTED_OAUTH_TOKEN | Server responded with unsupported OAuth token type. |
| WWV_RENDER_REPORT3.FOUND_BUT_NOT_DISPLAYED | Minimum row requested: %0, rows found but not displayed: %1 |
| WWV_RENDER_REPORT3.SORT_BY_THIS_COLUMN | Sort by this column. |
| WWV_RENDER_REPORT3.UNSAVED_DATA | This form contains unsaved changes. Press Ok to proceed without saving your changes. |
| WWV_RENDER_REPORT3.X_Y_OF_MORE_THAN_Z | row(s) %0 - %1 of more than %2 |
| WWV_RENDER_REPORT3.X_Y_OF_Z | row(s)%0 - %1 of %2 |
| WWV_RENDER_REPORT3.X_Y_OF_Z_2 | %0 - %1 of %2 |

Interactive Report Messages Requiring Translation

[Table 21-4](#) lists the interactive report messages that require translation.

Table 21-4 Interactive Report Messages Requiring Translation

| Message Name | English Text |
|----------------------------|---------------------|
| 4150_COLUMN_NUMBER | Column %0 |
| APEXIR_3D | 3D |
| APEXIR_ACTIONS | Actions |
| APEXIR_ACTIONS_MENU | Actions Menu |
| APEXIR_ADD | Add |
| APEXIR_ADD_FUNCTION | Add Function |
| APEXIR_ADD_GROUP_BY_COLUMN | Add Group By Column |
| APEXIR_ADD_PIVOT_COLUMN | Add Pivot Column |
| APEXIR_ADD_ROW_COLUMN | Add Row Column |
| APEXIR_ADD_SUBSCRIPTION | Add Subscription |
| APEXIR_AGG_AVG | Average |
| APEXIR_AGG_COUNT | Count |
| APEXIR_AGG_MAX | Maximum |
| APEXIR_AGG_MEDIAN | Median |
| APEXIR_AGG_MIN | Minimum |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|---------------------------------|---|
| APEXIR_AGG_MODE | Mode |
| APEXIR_AGG_SUM | Sum |
| APEXIR_AGGREGATE | Aggregate |
| APEXIR_AGGREGATE_DESCRIPTION | Aggregates are displayed after each control break and at the end of the report. |
| APEXIR_AGGREGATION | Aggregation |
| APEXIR_ALL | All |
| APEXIR_ALL_COLUMNS | All Columns |
| APEXIR_ALL_ROWS | All Rows |
| APEXIR_ALTERNATIVE | Alternative |
| APEXIR_ALTERNATIVE_DEFAULT_NAME | Alternative Default: %0 |
| APEXIR_AND | and |
| APEXIR_APPLY | Apply |
| APEXIR_AS_OF | As of %0 |
| APEXIR_ASCENDING | Ascending |
| APEXIR_AVERAGE_X | Average %0 |
| APEXIR_BETWEEN | between |
| APEXIR_BGCOLOR | Background Color |
| APEXIR_BLUE | blue |
| APEXIR_BOTTOM | Bottom |
| APEXIR_CALENDAR | Calendar |
| APEXIR_CANCEL | Cancel |
| APEXIR_CATEGORY | Category |
| APEXIR_CELL | Cell |
| APEXIR_CHART | Chart |
| APEXIR_CHART_INITIALIZING | Initializing... |
| APEXIR_CHART_LABEL_NOT_NULL | Chart label must be specified |
| APEXIR_CHART_MAX_ROW_CNT | The maximum row count for a chart query limits the number of rows in the base query, not the number of rows displayed. Your base query exceeds the maximum row count of %0. Please apply a filter to reduce the number of records in your base query. |
| APEXIR_CHART_TYPE | Chart Type |
| APEXIR_CHECK_ALL | Check All |
| APEXIR_CHOOSE_DOWNLOAD_FORMAT | Choose report download format |
| APEXIR_CLEAR | clear |
| APEXIR_COLUMN | Column |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-----------------------------------|--|
| APEXIR_COLUMN_FILTER | Filter... |
| APEXIR_COLUMN_HEADING_MENU | Column Heading Menu |
| APEXIR_COLUMN_INFO | Column Information |
| APEXIR_COLUMN_LABEL | Column Label |
| APEXIR_COLUMN_N | Column %0 |
| APEXIR_COLUMNS | Columns |
| APEXIR_COMPARISON_CONTAINS | contains |
| APEXIR_COMPARISON_DOESNOT_CONTAIN | does not contain |
| APEXIR_COMPARISON_IN | in |
| APEXIR_COMPARISON_IS_IN_LAST | is in the last |
| APEXIR_COMPARISON_IS_IN_NEXT | is in the next |
| APEXIR_COMPARISON_IS_NOT_NULL | is not null |
| APEXIR_COMPARISON_IS_NULL | is null |
| APEXIR_COMPARISON_ISNOT_IN_LAST | is not in the last |
| APEXIR_COMPARISON_ISNOT_IN_NEXT | is not in the next |
| APEXIR_COMPARISON_LIKE | like |
| APEXIR_COMPARISON_NOT_IN | not in |
| APEXIR_COMPARISON_NOT_LIKE | not like |
| APEXIR_COMPARISON_REGEXP_LIKE | matches regular expression |
| APEXIR_COMPUTATION | Computation |
| APEXIR_COMPUTATION_EXPRESSION | Computation Expression |
| APEXIR_COMPUTATION_FOOTER | Create a computation using column aliases. |
| APEXIR_COMPUTATION_FOOTER_E1 | (B+C)*100 |
| APEXIR_COMPUTATION_FOOTER_E2 | INITCAP(B) ', ' INITCAP(C) |
| APEXIR_COMPUTATION_FOOTER_E3 | CASE WHEN A = 10 THEN B + C ELSE B END |
| APEXIR_COMPUTE | Compute |
| APEXIR_CONTROL_BREAK | Control Break |
| APEXIR_CONTROL_BREAK_COLUMNS | Control Break Columns |
| APEXIR_CONTROL_BREAKS | Control Breaks |
| APEXIR_COUNT_DISTINCT | Count Distinct |
| APEXIR_COUNT_DISTINCT_X | Count Distinct |
| APEXIR_COUNT_X | Count %0 |
| APEXIR_DAILY | Daily |
| APEXIR_DATA | Data |
| APEXIR_DATA_AS_OF | Report data as of %0 minutes ago. |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|---------------------------------|---|
| APEXIR_DATE | Date |
| APEXIR_DAY | Day |
| APEXIR_DEFAULT | Default |
| APEXIR_DEFAULT_REPORT_TYPE | Default Report Type |
| APEXIR_DELETE | Delete |
| APEXIR_DELETE_CHECKED | Delete Checked |
| APEXIR_DELETE_CONFIRM | Would you like to delete these report settings? |
| APEXIR_DELETE_CONFIRM_JS_DIALOG | Would you like to perform this delete action? |
| APEXIR_DELETE_DEFAULT_REPORT | Delete Default Report |
| APEXIR_DELETE_REPORT | Delete Report |
| APEXIR_DESCENDING | Descending |
| APEXIR_DESCRIPTION | Description |
| APEXIR_DETAIL_VIEW | Single Row View |
| APEXIR_DIRECTION | Direction |
| APEXIR_DISABLE | Disable |
| APEXIR_DISABLED | Disabled |
| APEXIR_DISPLAY | Display |
| APEXIR_DISPLAY_IN_REPORT | Display in Report |
| APEXIR_DISPLAYED | Displayed |
| APEXIR_DISPLAYED_COLUMNS | Displayed Columns |
| APEXIR_DO_NOT_AGGREGATE | - Do not aggregate - |
| APEXIR_DO_NOT_DISPLAY | Do Not Display |
| APEXIR_DOWN | Down |
| APEXIR_DOWNLOAD | Download |
| APEXIR_DUPLICATE_PIVOT_COLUMN | Duplicate pivot column. Pivot column list must be unique. |
| APEXIR_EDIT | Edit |
| APEXIR_EDIT_ALTERNATIVE_DEFAULT | Edit Alternative Default |
| APEXIR_EDIT_CHART | Edit Chart Settings |
| APEXIR_EDIT_CHART2 | Edit Chart |
| APEXIR_EDIT_FILTER | Edit Filter |
| APEXIR_EDIT_GROUP_BY | Edit Group By |
| APEXIR_EDIT_HIGHLIGHT | Edit Highlight |
| APEXIR_EDIT_PIVOT | Edit Pivot |
| APEXIR_EMAIL | Email |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-------------------------------|---|
| APEXIR_EMAIL_ADDRESS | Email Address |
| APEXIR_EMAIL_BCC | Bcc |
| APEXIR_EMAIL_BODY | Body |
| APEXIR_EMAIL_CC | Cc |
| APEXIR_EMAIL_FREQUENCY | Frequency |
| APEXIR_EMAIL_NOT_CONFIGURED | Email has not been configured for this application. Please contact your administrator. |
| APEXIR_EMAIL_REQUIRED | Email Address must be specified. |
| APEXIR_EMAIL_SEE_ATTACHED | See attached. |
| APEXIR_EMAIL_SUBJECT | Subject |
| APEXIR_EMAIL_SUBJECT_REQUIRED | Email Subject must be specified. |
| APEXIR_EMAIL_TO | To |
| APEXIR_ENABLE | Enable |
| APEXIR_ENABLE_DISABLE_ALT | Enable/Disable |
| APEXIR_ENABLED | Enabled |
| APEXIR_ERROR | Error |
| APEXIR_EXAMPLES | Examples |
| APEXIR_EXAMPLES_WITH_COLON | Examples: |
| APEXIR_EXCLUDE_NULL | Exclude Null Values |
| APEXIR_EXPAND_COLLAPSE_ALT | Expand/Collapse |
| APEXIR_EXPRESSION | Expression |
| APEXIR_FILTER | Filter |
| APEXIR_FILTER_EXPRESSION | Filter Expression |
| APEXIR_FILTER_EXPR_TOO_LONG | The filter expression is too long. |
| APEXIR_FILTER_TYPE | Filter Type |
| APEXIR_FILTERS | Filters |
| APEXIR_FINDER_ALT | Select columns to search. |
| APEXIR_FLASHBACK | Flashback |
| APEXIR_FLASHBACK_DESCRIPTION | A flashback query enables you to view the data as it existed at a previous point in time. |
| APEXIR_FLASHBACK_ERROR_MSG | Unable to perform flashback request. |
| APEXIR_FORMAT | Format |
| APEXIR_FORMAT_MASK | Format Mask |
| APEXIR_FUNCTION | Function |
| APEXIR_FUNCTION_N | Function %0 |
| APEXIR_FUNCTIONS | Functions |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|------------------------------|---|
| APEXIR_FUNCTIONS_OPERATORS | Functions / Operators |
| APEXIR_GO | Go |
| APEXIR_GREEN | green |
| APEXIR_GROUP_BY | Group By |
| APEXIR_GROUP_BY_COL_NOT_NULL | Group by column must be specified |
| APEXIR_GROUP_BY_COLUMN | Group By Column |
| APEXIR_GROUP_BY_MAX_ROW_CNT | The maximum row count for a Group By query limits the number of rows in the base query, not the number of rows displayed. Your base query exceeds the maximum row count of %0. Please apply a filter to reduce the number of records in your base query. |
| APEXIR_GROUP_BY_SORT | Group By Sort |
| APEXIR_GROUP_BY_SORT_ORDER | Group By Sort Order |
| APEXIR_HCOLUMN | Horizontal Column |
| APEXIR_HELP | Help |
| APEXIR_HELP_01 | <p>An Interactive Report displays a predetermined set of columns. The report may be further customized with an initial filter clause, a default sort order, control breaks, highlighting, computations, aggregates and a chart. Each Interactive Report can then be further customized and the results can be viewed, or downloaded, and the report definition can be stored for later use.</p> <p><p/></p> <p>An Interactive Report can be customized in three ways: the search bar, actions menu and column heading menu.</p> |
| APEXIR_HELP_ACTIONS_MENU | The actions menu is used to customize the display of your Interactive Report. |
| APEXIR_HELP_AGGREGATE | <p>Aggregates are mathematical computations performed against a column. Aggregates are displayed after each control break and at the end of the report within the column they are defined.</p> <p><p/></p> <ul style="list-style-type: none"> Aggregation allows you to select a previously defined aggregation to edit. Function is the function to be performed (e.g. SUM, MIN). Column is used to select the column to apply the mathematical function to. Only numeric columns will be displayed. |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|---------------------------------|---|
| APEXIR_HELP_CHART | <p>You can include one chart per Interactive Report. Once defined, you can switch between the chart and report views using links below the search bar.</p> <p><p/></p> <ul style="list-style-type: none"> Chart Type identifies the chart type to include. Select from horizontal bar, vertical bar, pie or line. Label allows you to select the column to be used as the label. Axis Title for Label is the title that will display on the axis associated with the column selected for Label. This is not available for pie chart. Value allows you to select the column to be used as the value. If your function is a COUNT, a Value does not need to be selected. Axis Title for Value is the title that will display on the axis associated with the column selected for Value. This is not available for pie chart. Function is an optional function to be performed on the column selected for Value. |
| APEXIR_HELP_COLUMN_HEADING_MENU | <p>Clicking on any column heading exposes a column heading menu.</p> <p><p/></p> <ul style="list-style-type: none"> Sort Ascending icon sorts the report by the column in ascending order. Sort Descending icon sorts the report by the column in descending order. Hide Column hides the column. Break Column creates a break group on the column. This pulls the column out of the report as a master record. Column Information displays help text about the column, if available. Text Area is used to enter case insensitive search criteria (no need for wild cards). Entering a value will reduce the list of values at the bottom of the menu. You can then select a value from the bottom and the selected value will be created as a filter using '=' (e.g. column = 'ABC'). Alternatively, you can click the flashlight icon and the entered value will be created as a filter with the 'LIKE' modifier (e.g. column LIKE '%ABC%'). List of Unique Values contains the first 500 unique values that meet your filters. If the column is a date, a list of date ranges is displayed instead. If you select a value, a filter will be created using '=' (e.g. column = 'ABC'). |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|---------------------------|--|
| APEXIR_HELP_COMPUTE | <p>Computations allow you to add computed columns to your report. These can be mathematical computations (e.g. NBR_HOURS/24) or standard Oracle functions applied to existing columns (some have been displayed for example, others, like TO_DATE, can also be used).</p> <p><p/></p> <ul style="list-style-type: none"> Computation allows you to select a previously defined computation to edit. Column Heading is the column heading for the new column. Format Mask is an Oracle format mask to be applied against the column (e.g. S9999). Format Mask is an Oracle format mask to be applied against the column (e.g. S9999). Computation is the computation to be performed. Within the computation, columns are referenced using the aliases displayed. <p></p> <p><p/></p> <p>Below computation, the columns in your query are displayed with their associated alias. Clicking on the column name or alias will write them into the Computation. Next to Columns is a Keypad. These are simply shortcuts of commonly used keys. On the far right are Functions.</p> <p><p/></p> <p>An example computation to display Total Compensation is:</p> <p><p/></p> <pre>CASE WHEN A = 'SALES' THEN B + C ELSE B END</pre> <p>(where A is ORGANIZATION, B is SALARY and C is COMMISSION)</p> |
| APEXIR_HELP_CONTROL_BREAK | <p>Used to create a break group on one or several columns. This pulls the columns out of the Interactive Report and displays them as a master record.</p> |
| APEXIR_HELP_DETAIL_VIEW | <p>To view the details of a single row at a time, click the single row view icon on the row you want to view. If available, the single row view will always be the first column. Depending on the customization of the Interactive Report, the single row view may be the standard view or a custom page that may allow update.</p> |
| APEXIR_HELP_DOWNLOAD | <p>Allows the current result set to be downloaded. The download formats will differ depending upon your installation and report definition but may include CSV, XLS, PDF, or RTF.</p> |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-----------------------|---|
| APEXIR_HELP_FILTER | Used to add or modify the where clause on the query. You first select a column (it does not need to be one that is displayed), select from a list of standard Oracle operators (=, !=, not in, between), and enter an expression to compare against. The expression is case sensitive and you can use % as a wildcard (for example, STATE_NAME like A%). |
| APEXIR_HELP_FLASHBACK | Performs a flashback query to allow you to view the data as it existed at a previous point in time. The default amount of time that you can flashback is 3 hours (or 180 minutes) but the actual amount will differ per database. |
| APEXIR_HELP_FORMAT | <p>Format enable you to customize the display of the report. Format contains the following submenu:</p> <ul style="list-style-type: none"> Sort Control Break Highlight Compute Aggregate Chart Group By Pivot |
| APEXIR_HELP_GROUP_BY | You can define one Group By view per saved report. Once defined, you can switch between the group by and report views using view icons on the Search bar. To create a Group By view, you select: <p></p> <ul style="list-style-type: none"> the columns on which to group the columns to aggregate along with the function to be performed (average, sum, count, etc.) |
| APEXIR_HELP_HIGHLIGHT | Highlighting allows you to define a filter. The rows that meet the filter are highlighted using the characteristics associated with the filter. <p><p/></p> <ul style="list-style-type: none"> Name is used only for display. Sequence identifies the sequence in which the rules will be evaluated. Enabled identifies if the rule is enabled or disabled. Highlight Type identifies whether the Row or Cell should be highlighted. If Cell is selected, the column referenced in the Highlight Condition is highlighted. Background Color is the new color for the background of the highlighted area. Text Color is the new color for the text in the highlighted area. Highlight Condition defines your filter condition. |
| APEXIR_HELP_PIVOT | You can define one Pivot view per saved report. Once defined, you can switch between the pivot and report views using view icons on the Search bar. To create a Pivot view, you select: <p></p> <ul style="list-style-type: none"> the columns on which to pivot the columns to display as rows the columns to aggregate along with the function to be performed (average, sum, count, etc.) |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-----------------------------|--|
| APEXIR_HELP_REPORT_SETTINGS | <p>If you have customized your Interactive Report, the report settings will be displayed below the Search Bar and above the report. If you have saved customized reports, they will be shown as tabs. You can access your alternate views by clicking the tabs. Below the tabs are the report settings for the current report. This area can be collapsed and expanded using the icon on the left.</p> <p><p/></p> <p>For each report setting, you can:</p> <ul style="list-style-type: none"> Edit by clicking the name. Disable/Enable by unchecking or checking the Enable/Disable check box. This is used to temporarily turn off and on the setting. Remove by click the Remove icon. This permanently removes the setting. <p><p/></p> <p>If you have created a chart, you can toggle between the report and chart using the Report View and Chart View links shown on the right. If you are viewing the chart, you can also use the Edit Chart link to edit the chart settings.</p> |
| APEXIR_HELP_RESET | Resets the report back to the default settings, removing any customizations that you have made. |
| APEXIR_HELP_ROWS_PER_PAGE | Sets the number of records to display per page. |
| APEXIR_HELP_SAVE_REPORT | <p>Saves the customized report for future use. You provide a name and optional description and can make the report accessible to the public (that is, all users who can access the primary default report). You can save four types of interactive reports:</p> <ul style="list-style-type: none"> • Primary Default (Developer Only). The Primary Default is the report that initially displays. Primary Default reports cannot be renamed or deleted. • Alternative Report (Developer Only). Enables developers to create multiple report layouts. Only developers can save, rename, or delete an Alternative Report. • Public Report (End user). Can be saved, renamed, or deleted by the end user who created it. Other users can view and save the layout as another report. • Private Report (End user). Only the end user that created the report can view, save, rename or delete the report. <p>If you save customized reports, a Reports selector displays in the Search bar to the left of the Rows selector (if this feature is enabled).</p> |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-------------------------------------|--|
| APEXIR_HELP_SEARCH_BAR | <p>At the top of each report page is a search region. The region provides the following features:</p> <p><p/></p> <ul style="list-style-type: none"> Select columns icon allows you to identify which column to search (or all). Text area allows for case insensitive search criteria (no need for wild cards). Rows selects the number of records to display per page. [Go] button executes the search. Actions Menu icon displays the actions menu (discussed next). <p><p/></p> <p>Please note that all features may not be available for each report.</p> |
| APEXIR_HELP_SEARCH_BAR_ACTIONS_MENU | Actions Menu enables you to customize a report. See the sections that follow. |
| APEXIR_HELP_SEARCH_BAR_FINDER | Select columns icon enables you to identify which column to search (or all). |
| APEXIR_HELP_SEARCH_BAR_REPORTS | Reports displays alternate default and saved private or public reports. |
| APEXIR_HELP_SEARCH_BAR_ROWS | Rows sets the number of records to display per page. |
| APEXIR_HELP_SEARCH_BAR_TEXTBOX | Text area enables you to enter case insensitive search criteria (wild card characters are implied). Go button executes the search. Hitting the enter key will also execute the search when the cursor is in the search text area. |
| APEXIR_HELP_SEARCH_BAR_VIEW | View Icons switches between the icon, report, detail, chart, group by, and pivot views of the report if they are defined. |
| APEXIR_HELP_SELECT_COLUMNS | Used to modify the columns displayed. The columns on the right are displayed. The columns on the left are hidden. You can reorder the displayed columns using the arrows on the far right. Computed columns are prefixed with **. |
| APEXIR_HELP_SORT | Used to change the column(s) to sort on and whether to sort ascending or descending. You can also specify how to handle nulls (use the default setting, always display them last or always display them first). The resulting sorting is displayed to the right of column headings in the report. |
| APEXIR_HELP_SUBSCRIPTION | When you add a subscription, you provide an email address (or multiple email addresses, separated by commas), email subject, frequency, and start and end dates. The resulting emails include an HTML version of the interactive report containing the current data using the report setting that were present when the subscription was added. |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|--------------------------------|---|
| APEXIR_HIDE_COLUMN | Hide Column |
| APEXIR_HIGHLIGHT | Highlight |
| APEXIR_HIGHLIGHT_CONDITION | Highlight Condition |
| APEXIR_HIGHLIGHT_TYPE | Highlight Type |
| APEXIR_HIGHLIGHT_WHEN | Highlight When |
| APEXIR_HIGHLIGHTS | Highlights |
| APEXIR_INACTIVE_SETTING | 1 inactive setting |
| APEXIR_INACTIVE_SETTINGS | %0 inactive settings |
| APEXIR_INTERACTIVE_REPORT_HELP | Interactive Report Help |
| APEXIR_INVALID | Invalid |
| APEXIR_INVALID_COMPUTATION | Invalid computation expression. %0 |
| APEXIR_INVALID_END_DATE | The end date must be greater than the start date. |
| APEXIR_INVALID_END_DATE | The end date must be greater than the start date. |
| APEXIR_INVALID_FILTER | Invalid filter expression. %0 |
| APEXIR_INVALID_FILTER_QUERY | Invalid filter query |
| APEXIR_INVALID_SETTING | 1 invalid setting |
| APEXIR_INVALID_SETTINGS | %0 invalid settings |
| APEXIR_IS_IN_THE_LAST | %0 is in the last %1 |
| APEXIR_IS_IN_THE_NEXT | %0 is in the next %1 |
| APEXIR_IS_NOT_IN_THE_LAST | %0 is not in the last %1 |
| APEXIR_IS_NOT_IN_THE_NEXT | %0 is not in the next %1 |
| APEXIR_KEYPAD | Keypad |
| APEXIR_LABEL | Label |
| APEXIR_LABEL_AXIS_TITLE | Axis Title for Label |
| APEXIR_LABEL_PREFIX | Label Prefix |
| APEXIR_LAST_DAY | Last Day |
| APEXIR_LAST_HOUR | Last Hour |
| APEXIR_LAST_MONTH | Last Month |
| APEXIR_LAST_WEEK | Last Week |
| APEXIR_LAST_X_DAYS | Last %0 Days |
| APEXIR_LAST_X_HOURS | Last %0 Hours |
| APEXIR_LAST_X_YEARS | Last %0 Years |
| APEXIR_LAST_YEAR | Last Year |
| APEXIR_LINE | Line |
| APEXIR_MAP_IT | Map it |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|--|---|
| APEXIR_MAX_QUERY_COST | The query is estimated to exceed the maximum allowed resources. Please modify your report settings and try again. |
| APEXIR_MAX_ROW_CNT | This query returns more then %0 rows, please filter your data to ensure complete results. |
| APEXIR_MAX_X | Maximum %0 |
| APEXIR_MEDIAN_X | Median %0 |
| APEXIR_MIN_AGO | %0 minutes ago |
| APEXIR_MIN_X | Minimum %0 |
| APEXIR_MONTH | Month |
| APEXIR_MONTHLY | Monthly |
| APEXIR_MORE_DATA | More Data |
| APEXIR_MOVE | Move |
| APEXIR_MOVE_ALL | Move All |
| APEXIR_MULTIIIR_PAGE_REGION_STATIC_ID_REQUIRE D | Region Static ID must be specified as the page contains multiple interactive reports. |
| APEXIR_NAME | Name |
| APEXIR_NEW_AGGREGATION | New Aggregation |
| APEXIR_NEW_CATEGORY | - New Category - |
| APEXIR_NEW_CATEGORY_LABEL | New Category |
| APEXIR_NEW_COMPUTATION | New Computation |
| APEXIR_NEXT | > |
| APEXIR_NEXT_DAY | Next Day |
| APEXIR_NEXT_HOUR | Next Hour |
| APEXIR_NEXT_MONTH | Next Month |
| APEXIR_NEXT_WEEK | Next Week |
| APEXIR_NEXT_X_DAYS | Next %0 Days |
| APEXIR_NEXT_X_HOURS | Next %0 Hours |
| APEXIR_NEXT_X_YEARS | Next %0 Years |
| APEXIR_NEXT_YEAR | Next Year |
| APEXIR_NO | No |
| APEXIR_NO_AGGREGATION_DEFINED | No aggregation defined. |
| APEXIR_NO_COLUMN_INFO | No column information available. |
| APEXIR_NO_COMPUTATION_DEFINED | No computation defined. |
| APEXIR_NO_END_DATE | - No End Date - |
| APEXIR_NONE | - None - |
| APEXIR_NOT_VALID_EMAIL | Not a valid email address. |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|--|---|
| APEXIR_NULL_SORTING | Null Sorting |
| APEXIR_NULLS_ALWAYS_FIRST | Nulls Always First |
| APEXIR_NULLS_ALWAYS_LAST | Nulls Always Last |
| APEXIR_NUMERIC_FLASHBACK_TIME | Flashback time must be numeric. |
| APEXIR_NUMERIC_SEQUENCE | Sequence must be numeric. |
| APEXIR_OPERATOR | Operator |
| APEXIR_ORANGE | orange |
| APEXIR_OTHER | Other |
| APEXIR_PERCENT_OF_TOTAL_COUNT_X | Percent of Total Count %0 (%) |
| APEXIR_PERCENT_OF_TOTAL_SUM_X | Percent of Total Sum %0 (%) |
| APEXIR_PERCENT_TOTAL_COUNT | Percent of Total Count |
| APEXIR_PERCENT_TOTAL_SUM | Percent of Total Sum |
| APEXIR_PIE | Pie |
| APEXIR_PIVOT | Add Pivot Column |
| APEXIR_PIVOT_AGG_NOT_NULL | Aggregate must be specified. |
| APEXIR_PIVOT_AGG_NOT_ON_ROW_COL | You cannot aggregate on a column selected to as row column. |
| APEXIR_PIVOT_COLUMN_N | Pivot Column %0 |
| APEXIR_PIVOT_COLUMN_NOT_NULL | Pivot column must be specified. |
| APEXIR_PIVOT_COLUMNS | Pivot Columns |
| APEXIR_PIVOT_MAX_ROW_CNT | The maximum row count for a Pivot query limits the number of rows in the base query, not the number of rows displayed. Your base query exceeds the maximum row count of %0. Please apply a filter to reduce the number of records in your base query. |
| APEXIR_PIVOT_ROW_COLUMN_INVALID | Select different row column. The HTML expression or link in the row column contains column defined as pivot or aggregate column. |
| APEXIR_PIVOT_SORT | Pivot Sort |
| APEXIR_PREVIOUS | < |
| APEXIR_PRIMARY | Primary |
| APEXIR_PRIMARY_REPORT | Primary Report |
| APEXIR_PRIVATE | Private |
| APEXIR_PUBLIC | Public |
| APEXIR_RED | red |
| APEXIR_REGION_STATIC_ID_DOES_NOT_EXIST | Region Static ID %0 does not exist. |
| APEXIR_REMOVE | Remove |
| APEXIR_REMOVE_AGGREGATE | Remove Aggregate |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|--|---|
| APEXIR_REMOVE_ALL | Remove All |
| APEXIR_REMOVE_CHART | Remove Chart |
| APEXIR_REMOVE_CONTROL_BREAK | Remove Control Break |
| APEXIR_REMOVE_FILTER | Remove Filter |
| APEXIR_REMOVE_FLASHBACK | Remove Flashback |
| APEXIR_REMOVE_GROUP_BY | Remove Group By |
| APEXIR_REMOVE_HIGHLIGHT | Remove Highlight |
| APEXIR_REMOVE_PIVOT | Remove Pivot |
| APEXIR_REMOVE_REPORT | Remove Report |
| APEXIR_RENAME_DEFAULT_REPORT | Rename Default Report |
| APEXIR_RENAME_REPORT | Rename Report |
| APEXIR_REPORT | Report |
| APEXIR_REPORT_ALIAS_DOES_NOT_EXIST | Saved Interactive report with alias %0 does not exist. |
| APEXIR_REPORT_DISPLAY_COLUMN_LIMIT_REACHED | The number of display columns in the report reached the limit. Please click Select Columns under Actions menu to minimize the report display column list. |
| APEXIR_REPORT_DOES_NOT_EXIST | Report does not exist. |
| APEXIR_REPORT_ID_DOES_NOT_EXIST | Saved Interactive Report ID %0 does not exist. |
| APEXIR_REPORT_SETTINGS | Report Settings |
| APEXIR_REPORT_VIEW | < Report View |
| APEXIR_REPORTS | Reports |
| APEXIR_RESET | Reset |
| APEXIR_RESET_CONFIRM | Restore report to the default settings. |
| APEXIR_ROW | Row |
| APEXIR_ROW_COL_DIFF_FROM_PIVOT_COL | Row column must be different from the pivot column. |
| APEXIR_ROW_COLUMN_N | Row Column %0 |
| APEXIR_ROW_COLUMN_NOT_NULL | Row column must be specified. |
| APEXIR_ROW_COLUMNS | Row Columns |
| APEXIR_ROW_FILTER | Row Filter |
| APEXIR_ROW_OF | Row %0 of %1 |
| APEXIR_ROW_ORDER | Row Order |
| APEXIR_ROW_TEXT_CONTAINS | Row text contains |
| APEXIR_ROWS | Rows |
| APEXIR_ROWS_PER_PAGE | Rows Per Page |
| APEXIR_RPT_DISP_COL_EXCEED | The number of display columns in the report reached the limit. Please click Select Columns under Actions menu to minimize the report display column list. |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-------------------------------------|--|
| APEXIR_SAVE | Save |
| APEXIR_SAVE_AS_DEFAULT | Save as Default |
| APEXIR_SAVE_DEFAULT_CONFIRM | The current report settings are used as the default for all users. |
| APEXIR_SAVE_DEFAULT_REPORT | Save Default Report |
| APEXIR_SAVE_REPORT | Save Report |
| APEXIR_SAVE_REPORT_DEFAULT | Save Report * |
| APEXIR_SAVED_REPORT | Saved Report |
| APEXIR_SAVED_REPORT_MSG | Saved Report = "%0" |
| APEXIR_SEARCH | Search |
| APEXIR_SEARCH_BAR | Search Bar |
| APEXIR_SEARCH_COLUMN | Search: %0 |
| APEXIR_SEARCH_REPORT | Search Report |
| APEXIR_SELECT_CATEGORY | - Select Category - |
| APEXIR_SELECT_COLUMN | - Select Column - |
| APEXIR_SELECT_COLUMNS | Select Columns |
| APEXIR_SELECT_COLUMNS_FOOTER | Computed columns are prefixed with **. |
| APEXIR_SELECT_FUNCTION | - Select Function - |
| APEXIR_SELECT_GROUP_BY_COLUMN | - Select Group By Column - |
| APEXIR_SELECT_PIVOT_COLUMN | - Select Pivot Column - |
| APEXIR_SELECT_ROW | Select Row |
| APEXIR_SELECT_ROW_COLUMN | - Select Row Column - |
| APEXIR_SELECT_SORT_COLUMN | - Select Sort Column - |
| APEXIR_SELECT_VALUE | Select Value |
| APEXIR_SELECTED_COLUMNS | Selected Columns |
| APEXIR_SEND | Send |
| APEXIR_SEQUENCE | Sequence |
| APEXIR_SORT | Sort |
| APEXIR_SORT_ASCENDING | Sort Ascending |
| APEXIR_SORT_COLUMN | Sort Column |
| APEXIR_SORT_DESCENDING | Sort Descending |
| APEXIR_SORT_ORDER | Sort Order |
| APEXIR_SPACE_AS_IN_ONE_EMPTY_STRING | space |
| APEXIR_STATUS | Status |
| APEXIR_SUBSCRIPTION | Subscription |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-----------------------------------|--|
| APEXIR_SUBSCRIPTION_ENDING | Ending |
| APEXIR_SUBSCRIPTION_STARTING_FROM | Starting From |
| APEXIR_SUM_X | Sum %0 |
| APEXIR_TABLE_SUMMARY | Region = %0, Report = %1, View = %2, Displayed Rows Start = %3, Displayed Rows End = %4, Total Rows = %5 |
| APEXIR_TEXT_COLOR | Text Color |
| APEXIR_TIME_DAYS | days |
| APEXIR_TIME_HOURS | hours |
| APEXIR_TIME_MINS | minutes |
| APEXIR_TIME_MONTHS | months |
| APEXIR_TIME_WEEKS | weeks |
| APEXIR_TIME_YEARS | years |
| APEXIR_TOGGLE | Toggle |
| APEXIR_TOP | Top |
| APEXIR_UNAUTHORIZED | Unauthorized |
| APEXIR_UNGROUPED_COLUMN | Ungrouped Column |
| APEXIR_UNIQUE_COLUMN_HEADING | Column Heading must be unique. |
| APEXIR_UNIQUE_HIGHLIGHT_NAME | Highlight Name must be unique. |
| APEXIR_UNSUPPORTED_DATA_TYPE | unsupported data type |
| APEXIR_UP | Up |
| APEXIR_VALID_COLOR | Please enter a valid color. |
| APEXIR_VALID_FORMAT_MASK | Please enter a valid format mask. |
| APEXIR_VALUE | Value |
| APEXIR_VALUE_AXIS_TITLE | Axis Title for Value |
| APEXIR_VALUE_REQUIRED | Value Required |
| APEXIR_VCOLUMN | Vertical Column |
| APEXIR_VIEW_CHART | View Chart |
| APEXIR_VIEW_DETAIL | View Detail |
| APEXIR_VIEW_GROUP_BY | View Group By |
| APEXIR_VIEW_ICONS | View Icons |
| APEXIR_VIEW_PIVOT | View Pivot |
| APEXIR_VIEW_REPORT | View Report |
| APEXIR_WEEK | Week |
| APEXIR_WEEKLY | Weekly |
| APEXIR_WORKING_REPORT | Working Report |

Table 21-4 (Cont.) Interactive Report Messages Requiring Translation

| Message Name | English Text |
|-------------------------------|---|
| APEXIR_X_DAYS | %0 days |
| APEXIR_X_HOURS | %0 hours |
| APEXIR_X_MINS | %0 minutes |
| APEXIR_X_MONTHS | %0 months |
| APEXIR_X_WEEKS | %0 weeks |
| APEXIR_X_YEARS | %0 years |
| APEXIR_YEAR | Year |
| APEXIR_YELLOW | yellow |
| APEXIR_YES | Yes |
| IR_AS_DEFAULT_REPORT_SETTING | As Default Report Settings |
| IR_AS_NAMED_REPORT | As Named Report |
| IR_STAR | Only displayed for developers |
| LAYOUT.T.CONDITION_EXPR2 | Expression 2 |
| OUT_OF_RANGE | Invalid set of rows requested, the source data of the report has been modified. |
| REPORT | Report |
| REPORTING_PERIOD | Reporting Period |
| RESET | Reset Pagination |
| SAVED_REPORTS.PRIMARY.DEFAULT | Primary Default |
| WWV_RENDER_REPORT3.X_Y_OF_Z_2 | %0 - %1 of %2 |

Translating Data That Supports List of Values

You create a dynamic translation to translate dynamic pieces of data. For example, you might use a dynamic translation on a list of values based on a database query.

Dynamic translations differ from messages in that you query a specific string rather than a message name. You define dynamic translations on the Dynamic Translations page. You then use the `APEX_LANG.LANG` API to return the dynamic translation string identified by the `p_primary_text_string` parameter.

- [Defining a Dynamic Translation](#)
 You define dynamic translations on the Dynamic Translations page. A dynamic translation consists of a translate-from language string, a language code, and a translate-to string.

See Also:

`APEX_LANG` in *Oracle Application Express API Reference*

Defining a Dynamic Translation

You define dynamic translations on the Dynamic Translations page. A dynamic translation consists of a translate-from language string, a language code, and a translate-to string.

To define a dynamic translation:

1. Navigate to the Translate Application page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. Under Globalization, click **Translate Application**.
2. On the Translate Application page under Translation Utilities, select **Dynamic Translations**.
3. On the Dynamic Translations page, click **Create** and specify the following:
 - a. **Language** - Select a target language.
 - b. **Translate From Text** - Enter the source text to be translated.
 - c. **Translate To Text** - Enter the translated text.
4. Click **Create**.

Understanding Supported Globalization Codes

If you are building a multilingual application, it is important to understand how globalization codes affect the way in which your application runs. These codes are set automatically based on the application-level Globalization attributes you select.

- [Oracle Application Express Globalization Codes](#)
NLS_LANGUAGE and NLS_TERRITORY determine the default presentation of numbers, dates, and currencies.



See Also:

["Specifying the Primary Language for an Application"](#)

Oracle Application Express Globalization Codes

NLS_LANGUAGE and NLS_TERRITORY determine the default presentation of numbers, dates, and currencies.

Table 21-5 describes the globalization codes in Oracle Application Express.

Table 21-5 Oracle Application Express Globalization Codes

| Language Name | Language Code | NLS_LANGUAGE | NLS_TERRITORY |
|-------------------------|---------------|---------------------|----------------------|
| Afrikaans | af | ENGLISH | SOUTH AFRICA |
| Arabic | ar | ARABIC | UNITED ARAB EMIRATES |
| Arabic (Algeria) | ar-dz | ARABIC | ALGERIA |
| Arabic (Bahrain) | ar-bh | ARABIC | BAHRAIN |
| Arabic (Egypt) | ar-eg | EGYPTIAN | EGYPT |
| Arabic (Iraq) | ar-iq | ARABIC | IRAQ |
| Arabic (Jordan) | ar-jo | ARABIC | JORDAN |
| Arabic (Kuwait) | ar-kw | ARABIC | KUWAIT |
| Arabic (Lebanon) | ar-lb | ARABIC | LEBANNON |
| Arabic (Libya) | ar-ly | ARABIC | LIBYA |
| Arabic (Morocco) | ar-ma | ARABIC | MOROCCO |
| Arabic (Oman) | ar-om | ARABIC | OMAN |
| Arabic (Qatar) | ar-qa | ARABIC | QATAR |
| Arabic (Saudi Arabia) | ar-sa | ARABIC | SAUDI ARABIA |
| Arabic (Syria) | ar-sy | ARABIC | SYRIA |
| Arabic (Tunisia) | ar-tn | ARABIC | TUNISIA |
| Arabic (U.A.E.) | ar-ae | ARABIC | UNITED ARAB EMIRATES |
| Arabic (YEMEN) | ar-ye | ARABIC | YEMEN |
| Assamese | as | ASSAMESE | INDIA |
| Basque | eu | FRENCH | FRANCE |
| Belarusian | be | RUSSIAN | RUSSIA |
| Bengali | bn | BANGLA | BANGLADESH |
| Bulgarian | bg | BULGARIAN | BULGARIA |
| Catalan | ca | CATALAN | CATALONIA |
| Chinese | zh | SIMPLIFIED CHINESE | CHINA |
| Chinese (China) | zh-cn | SIMPLIFIED CHINESE | CHINA |
| Chinese (Hong Kong SAR) | zh-hk | TRADITIONAL CHINESE | HONG KONG |
| Chinese (Macau SAR) | zh-mo | TRADITIONAL CHINESE | HONG KONG |
| Chinese (Singapore) | zh-sg | SIMPLIFIED CHINESE | SINGAPORE |
| Chinese (Taiwan) | zh-tw | TRADITIONAL CHINESE | TAIWAN |
| Croatian | hr | CROATIAN | CROATIA |
| Czech | cs | CZECH | CZECH REPUBLIC |
| Danish | da | DANISH | DENMARK |
| Dutch (Belgium) | nl-be | DUTCH | BELGIUM |

Table 21-5 (Cont.) Oracle Application Express Globalization Codes

| Language Name | Language Code | NLS_LANGUAGE | NLS_TERRITORY |
|--------------------------|---------------|-----------------|-----------------|
| Dutch (Netherlands) | nl | DUTCH | THE NETHERLANDS |
| English | en | AMERICAN | AMERICA |
| English (Australia) | en-au | ENGLISH | AUSTRALIA |
| English (Belize) | en-bz | ENGLISH | UNITED KINGDOM |
| English (Canada) | en-ca | ENGLISH | CANADA |
| English (Ireland) | en-ie | ENGLISH | IRELAND |
| English (Jamaica) | en-jm | ENGLISH | UNITED KINGDOM |
| English (New Zealand) | en-nz | ENGLISH | NEW ZEALAND |
| English (Philippines) | en-ph | ENGLISH | PHILIPPINES |
| English (South Africa) | en-za | ENGLISH | SOUTH AFRICA |
| English (Trinidad) | en-tt | ENGLISH | UNITED KINGDOM |
| English (United Kingdom) | en-gb | ENGLISH | UNITED KINGDOM |
| English (United States) | en-us | AMERICAN | AMERICA |
| English (Zimbabwe) | en-zw | ENGLISH | UNITED KINGDOM |
| Estonian | et | ESTONIAN | ESTONIA |
| Faeroese | fo | ENGLISH | UNITED KINGDOM |
| Farsi | fa | ENGLISH | UNITED KINGDOM |
| Finnish | fi | FINNISH | FINLAND |
| French (Belgium) | fr-be | FRENCH | BELGIUM |
| French (Canada) | fr-ca | CANADIAN FRENCH | CANADA |
| French (France) | fr | FRENCH | FRANCE |
| French (Luxembourg) | fr-lu | FRENCH | LUXEMBOURG |
| French (Monaco) | fr-mc | FRENCH | FRANCE |
| French (Switzerland) | fr-ch | FRANCH | SWITZERLAND |
| FYRO Macedonian | mk | MACEDONIAN | FYR MACEDONIA |
| Gaelic | gd | ENGLISH | UNITED KINGDOM |
| Galician | gl | SPANISH | SPAIN |
| German (Austria) | de-at | GERMAN | AUSTRIA |
| German (Germany) | de | GERMAN | GERMANY |
| German (Liechtenstein) | de-li | GERMAN | GERMANY |
| German (Luxembourg) | de-lu | GERMAN | LUXEMBOURG |
| German (Switzerland) | de-ch | GERMAN | SWITZERLAND |
| Greek | el | GREEK | GREECE |
| Gujarati | gu | GUJARATI | INDIA |

Table 21-5 (Cont.) Oracle Application Express Globalization Codes

| Language Name | Language Code | NLS_LANGUAGE | NLS_TERRITORY |
|-----------------------|---------------|----------------------|-----------------------|
| Hebrew | he | HEBREW | ISRAEL |
| Hindi | hi | HINDI | INDIA |
| Hungarian | hu | HUNGARIAN | HUNGARY |
| Icelandic | is | ICELANDIC | ICELAND |
| Indonesian | id | INDONESIAN | INDONESIA |
| Italian (Italy) | it | ITALIAN | ITALY |
| Italian (Switzerland) | it-ch | ITALIAN | SWITZERLAND |
| Japanese | ja | JAPANESE | JAPAN |
| Kannada | kn | KANNADA | INDIA |
| Kazakh | kk | CYRILLIC KAZAKH | KAZAKHSTAN |
| Konkani | kok | KOREAN | KOREA |
| Korean | ko | KOREAN | KOREA |
| Kyrgyz | kz | RUSSIAN | RUSSIA |
| Latvian | lv | LATVIAN | LATVIA |
| Lithuanian | lt | LITHUANIAN | LITHUANIANA |
| Malay (Malaysia) | ms | MALAY | MALAYSIA |
| Malayalam | ml | MALAYALAM | INDIA |
| Maltese | mt | ENGLISH | UNITED KINGDOM |
| Marathi | mr | ENGLISH | INDIA |
| Nepali (India) | ne | ENGLISH | UNITED KINGDOM |
| Norwegian (Bokmal) | nb-no | NORWEGIAN | NORWAY |
| Norwegian (Bokmal) | no | NORWEGIAN | NORWAY |
| Norwegian (Nynorsk) | nn-no | NORWEGIAN | NORWAY |
| Oriya | or | ORIYA | INDIA |
| Polish | pl | POLISH | POLAND |
| Portuguese (Brazil) | pt-br | BRAZILIAN PORTUGUESE | BRAZIL |
| Portuguese (Portugal) | pt | PORTUGUESE | PORTUGAL |
| Punjabi | pa | PUNJABI | INDIA |
| Romanian | ro | ROMANIAN | ROMANIA |
| Russian | ru | RUSSIAN | RUSSIA |
| Russian (Moldova) | ru-md | RUSSIAN | RUSSIA |
| Serbia | sr | CYRILLIC SERBIAN | SERBIA AND MONTENEGRO |
| Slovak | sk | SLOVAK | SLOVAKIA |
| Slovenian | sl | SLOVENIAN | SLOVENIA |

Table 21-5 (Cont.) Oracle Application Express Globalization Codes

| Language Name | Language Code | NLS_LANGUAGE | NLS_TERRITORY |
|------------------------------|---------------|------------------------|----------------|
| Spanish (Argentina) | es-ar | LATIN AMERICAN SPANISH | ARGENTINA |
| Spanish (Bolivia) | es-bo | LATIN AMERICAN SPANISH | ARGENTINA |
| Spanish (Chile) | es-cl | LATIN AMERICAN SPANISH | CHILE |
| Spanish (Columbia) | ec-co | LATIN AMERICAN SPANISH | COLUMBIA |
| Spanish (Costa Rica) | es-cr | LATIN AMERICAN SPANISH | COSTA RICA |
| Spanish (Dominican Republic) | es-do | LATIN AMERICAN SPANISH | PUERTO RICO |
| Spanish (Ecuador) | es-ec | LATIN AMERICAN SPANISH | ECUDOR |
| Spanish (El Salvador) | es-sv | LATIN AMERICAN SPANISH | EL SALVADOR |
| Spanish (Guatemala) | es-gt | LATIN AMERICAN SPANISH | GUATEMALA |
| Spanish (Honduras) | es-hn | LATIN AMERICAN SPANISH | GUATEMALA |
| Spanish (Mexico) | es-mx | MEXICAN SPANISH | MEXICO |
| Spanish (Nicaragua) | es-ni | LATIN AMERICAN SPANISH | Nicaragua |
| Spanish (Panama) | es-pa | LATIN AMERICAN SPANISH | Panama |
| Spanish (Paraguay) | es-py | LATIN AMERICAN SPANISH | ARGENTINA |
| Spanish (Peru) | es-pe | LATIN AMERICAN SPANISH | PERU |
| Spanish (Puerto Rico) | es-pr | LATIN AMERICAN SPANISH | PUERTO RICO |
| Spanish (Traditional Sort) | es | LATIN AMERICAN SPANISH | SPAIN |
| Spanish (United States) | es-us | LATIN AMERICAN SPANISH | AMERICAN |
| Spanish (Uruguay) | es-uy | LATIN AMERICAN SPANISH | ARGENTINA |
| Spanish (Venezuela) | es-ve | LATIN AMERICAN SPANISH | VENEZUELA |
| Swedish | sv | SWEDISH | SWEDEN |
| Swedish | sv-fi | SWEDISH | FINLAND |
| Tamil | ta | TAMIL | INDIA |
| Telugu | te | TELUGU | INDIA |
| Thai | th | THAI | THAILAND |
| Turkish | tr | TURKISH | TURKEY |
| Ukrainian | uk | UKRAINIAN | UKRAINE |
| Urdu | ur | ENGLISH | UNITED KINGDOM |
| Uzbek | uz | LATIN UZBEK | UZBEKISTAN |
| Vietnamese | vi | VIETNAMESE | VIETNAM |

Managing Application Performance

Learn about how to improve application performance.

This section describes how to improve application performance.

- [About Best Practices to Improve Performance](#)
Improve performance by using bind variables and including a #TIMING# substitution string in the region footer.
- [Identifying Performance Issues](#)
Consider the following key strategies to help identify and resolve performance issues with your Oracle Application Express application.
- [Utilizing Logs and Reports](#)
Review application logs and reports to help identify problematic pages.
- [About Utilizing Database Reporting](#)
Improve application performance by utilizing database reporting.
- [About Database Parameters that Impact Performance](#)
Evaluate whether or not database parameters are impacting performance.
- [About Limiting Resources](#)
Use Resource Manager to limit the types of system resources available to each user request.
- [About Uploading Static Files to Your Web Server](#)
To improve performance you should place static files on the Web server.
- [Creating Custom Activity Reports Using APEX_ACTIVITY_LOG](#)
Learn how to use the APEX_ACTIVITY_LOG view to query activity for the current workspace.



See Also:

["Debugging an Application"](#)

About Best Practices to Improve Performance

Improve performance by using bind variables and including a #TIMING# substitution string in the region footer.

For applications having a large number of concurrent users, maintaining optimal performance is critical. Best practices to consider include:

- Use bind variables within your application whenever possible. You can reference session state values using bind variable syntax in SQL queries and application logic such as PL/SQL executed from processes and validations. Accessing

session state using bind variables is the most efficient way to reference session state.

- Include a #TIMING# substitution string in the region footer so that you can view the timing of each region.

 **See Also:**

- ["Referencing Session State Using Bind Variable Syntax"](#)
- ["Using Substitution Strings"](#)

Identifying Performance Issues

Consider the following key strategies to help identify and resolve performance issues with your Oracle Application Express application.

- [About Troubleshooting Slow Running Applications](#)
- [Diagnosing Performance Issues](#)
- [About Debugging Problematic SQL Queries](#)
- [About Addressing Slow Queries](#)

About Troubleshooting Slow Running Applications

The first step in troubleshooting a slow running application is to determine the location of the bottleneck. You can evaluate application performance by reviewing the time it takes for the database to return a specific page request or submission. If this time is shorter than the time it takes to render the page on the user's browser, then other components are likely causing the perception that the application is slow. As a best practice, always check all of the components involved including client machines, the network, the middle-tier, the database, and disk and storage devices. Regarding the database, check the System Global Area (SGA), CPU utilization, locks, and database parameters. It is also important to determine if anything else is running within the database such as legacy applications, data warehouse, batch programs, and database jobs.

The majority of the time, performance issues with Oracle Application Express applications are due to developer-authored SQL and PL/SQL. You should utilize standard database performance tuning techniques to address these issues. Performance issues are rarely related to a bug with Oracle Application Express or the Oracle database.

 **See Also:**

Oracle Database Performance Tuning Guide

Diagnosing Performance Issues

To diagnose performance issues, follow these steps:

1. Identify the time window.
2. Isolate the largest consumers of database time.
3. Identify the Oracle Application Express workspace and application.
4. Correlate the offending SQL with a specific location within the application.

Once you have identified the slow running pages, you should run the application in debug mode to identify the specific components consuming the most time.

About Debugging Problematic SQL Queries

If your query does not seem to be running correctly, try running it in SQL*Plus, SQL Developer, or in SQL Commands. Any of these approaches will test your query outside the context of your application, making it easier to define the problem.

About Addressing Slow Queries

For optimized queries that still take more than a second to process, consider implementing a progress bar and preventing multiple page submissions. Users expect instant results and often resubmit a page if it does not return immediately. Unless managed properly, each page submission instigates a new session in the database and continues until completion. Multiple page submissions often degrade performance and may lead to database locks. This is especially true with interactive reports. As an alternative approach, try using Oracle Application Express collections to hold query results. By using collections, expensive and time consuming queries are only executed once for each user rather than each time they perform pagination or create a filter.

Tip:

You control whether end users may submit the same page more than once by configuring the page attribute, **Advanced, Enable duplicate page submissions**.

See Also:

- ["Editing Page Attributes "](#)
- ["About Using Collections"](#)

Utilizing Logs and Reports

Review application logs and reports to help identify problematic pages.

The Application Express engine records activity logs to capture usage and performance data. This information displays in various reports. If a particular page is accessed extensively or a page renders slowly then these reports can help you identify problematic pages.

You can then further analyze the problematic pages. For example, you can run a report query in SQL Developer to review the Explain Plan.

 **Tip:**

Another method of determining expensive elements within a page or process is to review the Application Express Debug mode.

- [How to Manage Log Files](#)
An Instance administrator can delete activity log files and manage the interval at which log file information is retained.
- [Viewing the Application Dashboard Report](#)
Learn how to view the Application Dashboard report.
- [Workspace Activity Reports](#)
Workspace administrators are users who perform administrator tasks specific to a workspace and have the access to various types of activity reports.
- [Instance Activity Reports](#)
Instance administrators are superusers that manage an entire hosted instance using the Application Express Administration Services application.

 **See Also:**

- "APEX_DEBUG" in *Oracle Application Express API Reference*
- "Utilizing Debug Mode"

How to Manage Log Files

An Instance administrator can delete activity log files and manage the interval at which log file information is retained.

If you wish to keep the activity log files indefinitely, you must create a batch job to copy records from the activity log table to another table.

 **See Also:**

"Managing Logs and Files" in *Oracle Application Express Administration Guide*

Viewing the Application Dashboard Report

Learn how to view the Application Dashboard report.

To view the Application Dashboard:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

The Application home page appears.

4. Click **Utilities**.
5. Click **Application Dashboard**.

The Application Dashboard appears. The first two regions on the left, Application Overview, and Pages by Types, list detailed information and statistics about the current application. The remaining regions are divided into the following sections:

- **Security** lists the current authentication scheme, number of public and non public pages, and the number of authorization schemes used within the current application.
 - **Templates** contains links to reports of templates used within the current application.
 - **Pages by Type** lists pages by type.
 - **Application Components** contains links to reports of application-level controls and logic, including lists of values, tabs, lists, application items, application processes, and application computations.
 - **Page Components** contains links to reports of page-level controls and logic, including items, buttons, processes, regions, branches, validations, and dynamic actions.
6. To view a specific report, click the number to the right of the component type or template.

Workspace Activity Reports

Workspace administrators are users who perform administrator tasks specific to a workspace and have the access to various types of activity reports.

Workspace administrators can access the following activity reports:

- **Page Views** - Contains reports of page views organized by view, user, application, application and page, day, hour, and by interactive report.
- **Developer Activity** - Offers reports of developer activity organized by developer, day, application, application changes, and day or month.
- **Page View Analysis** - Contains reports analyzing page views, such most viewed pages, page views by day, usage by day (chart), weighted page performance, and Websheet page views.
- **Sessions** - Lists active sessions with the current workspace (report or chart).

- **Login Attempts** - Offers reports listing login attempts, login attempts by authentication result, and a developer login summary.
- **Environment** - Contains reports of environments organized by user agent, browser, external clicks, or operating system.
- **Application Errors** - Contains a report of application errors.
- **Workspace Schema Reports** - Offers summaries of schema tablespace utilization and database privileges by schema, workspace schemas, and report tablespace utilization.



See Also:

"Monitoring Activity Within a Workspace" in *Oracle Application Express Administration Guide*

Instance Activity Reports

Instance administrators are superusers that manage an entire hosted instance using the Application Express Administration Services application.

Instance administrators can access the following activity reports:

- **Page Views** - View activity by application, user, workspace, day, or REST access.
- **Workspace Purge** - View a dashboard summary, inactive workspaces, workspaces purged, workspaces that became active, or a workspace purge log.
- **Environment Reports** - View a summary of used operating systems, browser types, user agent, or external sites.
- **Calendar Reports** - View workspaces by date last used, page views by day and then by application and user, or by hour.
- **Service Requests** - View new service requests or sign up survey activity.
- **Logs** - View the mail log, jobs log, automatic delete log, or monitor productivity or sample application installations.
- **Login Attempts** - View login attempts or developer last login.
- **Developer Activity** - View application changes by developer or workspace.



See Also:

"Monitoring Activity Across a Development Instance" in *Oracle Application Express Administration Guide* and "[About Oracle Application Express Administrator Roles](#)"

About Utilizing Database Reporting

Improve application performance by utilizing database reporting.

Oracle Application Express establishes a physical connection from the database pool as `APEX_PUBLIC_USER`, which calls the Application Express engine. The Application Express engine utilizes `SQL.DBMS_SYS_SQL` to parse SQL as another user, the parsing schema. Database `V$SESSION` records contain useful information for each SQL executed including Oracle Application Express specific information for the client information, client identifier, and module. You can use this information to identify the associated SQL when you generate Oracle Trace files and use `TKPROF` to analyze them. You can also add `&p_trace=YES` to the end of the Application Express URL or use `ALTER SESSION`.

Automatic Workload Repository (AWR) reports collect performance statistics every hour, by default. Active Session History (ASH) reports are a system-wide record of database activity. You can use these reports to identify resource intensive SQL statements. For PL/SQL packages, procedures and functions that in turn call other PL/SQL programs, you can use PL/SQL hierarchical profiler in Oracle Database 11g. These subprograms account for execution times separately.

Oracle Application Express also includes views which catalog everything related to Oracle Application Express. You can use these views to access information from Oracle Application Express using tools such as SQL Developer, SQL*Plus and SQL Commands (Oracle Application Express SQL Workshop). If you are granted the `APEX_ADMINISTRATOR_READ_ROLE` or `APEX_ADMINISTRATOR_ROLE` then you can query across the entire instance, rather than just the workspace your schema user is associated with. You should grant `APEX_ADMINISTRATOR_READ_ROLE` to monitoring users, while `APEX_ADMINISTRATOR_ROLE` should be used for instance administrators, who manage instance parameters, workspaces, and so on.

See Also:

- ["Enabling SQL Tracing and Using TKPROF"](#)
- *Oracle Database Performance Tuning Guide*
- *Oracle Application Express SQL Workshop Guide*

About Database Parameters that Impact Performance

Evaluate whether or not database parameters are impacting performance.

There are certain database parameters that can impact performance. In particular, you should review the memory related parameters (`MEMORY_TARGET`, `SGA_TARGET`, `PGA_AGGREGATE_TARGET`, `SHARED_POOL_SIZE`) parameters to ensure they are sized correctly. If you are using the Embedded PL/SQL Gateway, you should make sure that `SHARED_SERVERS` is sized correctly, to improve performance for multiple concurrent requests. You should also check that the database System Global Area (SGA) fits in real memory. An undersized SGA can dramatically impact Application Express performance. There needs to be sufficient memory in the Shared Pool for PL/SQL and in Buffer Cache for the Application Express metadata. If the SGA is sized correctly it should not be necessary to pin PL/SQL programs from the Application Express engine as they are accessed regularly and should not be swapped out of memory. Because the Application Express engine resides in the database, it is strongly recommended that you gather statistics on the `APEX_180200` schema. Accurate statistics allow the

cost-based optimizer to develop better query plans for accessing the application metadata.

About Limiting Resources

Use Resource Manager to limit the types of system resources available to each user request.

Database Resource Manager, an Oracle Database Enterprise Edition option, is exceedingly useful for maintaining an Oracle Application Express instance. Using Resource Manager you can limit the types of system resources (`CPU_PER_CALL` and `LOGICAL_READS_PER_CALL`) available to each user request. By setting up multiple resource profiles, with automatic switches between the profiles, as a user request exceeds a certain threshold you can reduce the CPU available to that transaction. For example you may define a default profile `APEX_HIGH` which can utilize 70% of CPU for 10 seconds, `APEX_MEDIUM` with 8% CPU for 120 seconds, `APEX_LOW` with 2% CPU for 1800 seconds, and `OTHERS` with 20% CPU. When a specific transaction exceeds 10 seconds, it is switched to the `APEX_MEDIUM` profile which drastically reduces the CPU available. If the transaction switches to `APEX_LOW` and exceeds 1800 seconds then you can cancel the SQL or kill the session. This approach ensures that one specific user or poorly written application cannot monopolize the CPU causing other users and applications to respond poorly.

Tip:

Instance administrators can define Resource Manager profiles for individual workspaces.

See Also:

- "Configuring Workspace Isolation Attributes" and "Configuring Instance-Level Workspace Isolation Attributes" in *Oracle Application Express Administration Guide*
- "Managing Resources with Oracle Database Resource Manager" in *Oracle Database Administrator's Guide*

About Uploading Static Files to Your Web Server

To improve performance you should place static files on the Web server.

As a best practice, you should also enable gzip compression on the Web server so that the file size is minimized when transmitting the static files to the browser. Furthermore, you should also enable file caching for browsers by sending expiration headers.

 **See Also:**

- ["Managing Static Application Files"](#)
- ["Managing Static Workspace Files"](#)

Creating Custom Activity Reports Using APEX_ACTIVITY_LOG

Learn how to use the APEX_ACTIVITY_LOG view to query activity for the current workspace.

- [Enabling Logging for an Application](#)
You enable logging for an application on the Edit Application Definition page.
- [APEX_ACTIVITY_LOG](#)
Use APEX_ACTIVITY_LOG to view and query all activity for the current workspace.
- [APEX_ACTIVITY_LOG Sample Query](#)

Enabling Logging for an Application

You enable logging for an application on the Edit Application Definition page.

To edit the application definition:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Edit Application Properties** button to the right of the application name.
The Edit Application Definition page appears.
4. Under Properties, locate Logging. Select **Yes** or **No**.

When set to **Yes**, every page view is logged, enabling an administrator to monitor user activity for each application. Disabling logging may be advisable for high volume applications.

 **Tip:**

You can only modify the Logging attribute if the Application Activity Logging attribute in Oracle Application Express Administration Services is set to **Use Application Setting**.

**See Also:**

"Enabling Application Activity" in Logging in *Oracle Application Express Administration Guide*

APEX_ACTIVITY_LOG

Use APEX_ACTIVITY_LOG to view and query all activity for the current workspace.

The APEX_ACTIVITY_LOG view records all activity in a workspace, including developer activity and application runtime activity. You can use APEX_ACTIVITY_LOG to view and query all activity for the current workspace. For example, you can use this view to develop monitoring reports within a specific application to provide real-time performance statistics.

Table 22-1 describes the columns in the APEX_ACTIVITY_LOG view.

Table 22-1 Columns in APEX_ACTIVITY_LOG

| Column | Type | Description |
|------------------------|----------------|---|
| time_stamp | DATE | Date and time that activity was logged at the end of the page view. |
| component_type | VARCHAR2(255) | Reserved for future use. |
| component_name | VARCHAR2(255) | Reserved for future use. |
| component_attribute | VARCHAR2(4000) | Title of page. |
| information | VARCHAR2(4000) | Reserved for future use. |
| elap | NUMBER | Elapsed time of page view in seconds. |
| num_rows | NUMBER | Number of rows processed on page. |
| userid | VARCHAR2(255) | User ID performing page view. |
| ip_address | VARCHAR2(4000) | IP address of client. |
| ir_report_id | NUMBER | Interactive report ID |
| ir_search | VARCHAR2 | Interactive report search criteria entered by users. |
| user_agent | VARCHAR2(4000) | web browser user agent of client. |
| flow_id | NUMBER | Application ID. |
| step_id | NUMBER | Page number. |
| session_id | NUMBER | Oracle Application Express session identifier. |
| sqlerrm | VARCHAR2(4000) | SQL Error message. |
| sqlerrm_component_type | VARCHAR2(255) | Reserved for future use. |
| sqlerrm_component_name | VARCHAR2(255) | Reserved for future use. |

To conserve space in the activity log, only the first log entry of each unique session contains the IP address and web browser user agent.

APEX_ACTIVITY_LOG Sample Query

The following example demonstrates how to create a report that displays the total number of page views and the average page view time in the past 24 hours for application 9529, and grouped by userid:

```
SELECT COUNT(*), AVG(elap), userid
       FROM APEX_ACTIVITY_LOG
      WHERE time_stamp > (SYSDATE-1)
            AND flow_id = 9529
      GROUP BY userid
```

Keep in mind that activity logging in an Oracle Application Express instance is rotated between two different log tables. Because there are two log tables, logging information is only as current as the oldest available entry in the logs. To persist your application specific log information for all time, you must either copy the log information into your own application table or implement logging directly in your application.

23

Debugging an Application

Learn about best practices for debugging an Oracle Application Express application.

- [Running Advisor to Check Application Integrity](#)
Use Oracle Application Express Advisor (Advisor) to check the integrity and quality of your Oracle Application Express application.
- [Reviewing Session State](#)
View current session state for your application by clicking the Session link on the Runtime Developer Toolbar.
- [About Monitoring Application and Page Resource Use](#)
Query the `V$SESSION` and `V$SQLAREA` views to monitor application and page resource use..
- [Enabling SQL Tracing and Using TKPROF](#)
Enable SQL tracing and then analyze the temporary file Oracle Application Express creates using the TKPROF utility.
- [Utilizing Debug Mode](#)
Use Debug mode to track down unexpected application behavior.
- [About Removing Controls and Components to Isolate a Problem](#)
If you have problems running a page, try removing controls and components one at a time.

See Also:

- "APEX_DEBUG" in *Oracle Application Express API Reference*
- "Managing Application Performance"

Running Advisor to Check Application Integrity

Use Oracle Application Express Advisor (Advisor) to check the integrity and quality of your Oracle Application Express application.

Oracle Application Express Advisor functions like a compiler or LINT and flags suspicious behavior or errors. Running Advisor checks the integrity of your application based on the underlying metadata.

- [About Oracle Application Express Advisor](#)
- [Running Advisor on an Entire Application](#)
- [Running Advisor on a Single Page](#)

About Oracle Application Express Advisor

Before deploying your application, you can use the Oracle Application Express Advisor to perform various sanity checks on your application. The Advisor checks for errors, security issues, performance bottlenecks, quality assurance, and other best practices.

 **Tip:**

Oracle Application Express Advisor includes checks specific to ensuring application accessibility. To learn more, see *Oracle Application Express Accessibility Guide*.

Running Advisor on an Entire Application

To run Advisor on an entire application:

1. Navigate to the appropriate application:
 - a. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
 - b. Select the application.
The Application home page appears.
2. Click **Utilities**.
3. Click **Advisor**.
The Advisor page appears.
4. Under Checks to Perform:
 - a. Expand Checks to Perform.
 - b. Review the selected options. Select and deselect options as appropriate.
5. In Pages(s), enter a comma separated list of pages in the field provided. To check all pages, leave this option blank.
6. Click **Perform Check** at the top of the page.
A Results page appears.

 **Tip:**

Current Advisor settings are used next time the check is performed.

7. To alter the existing preferences:
 - a. Under Filter Result, deselect the appropriate options.
 - b. Click **Apply Filter**.
8. To run Advisor again, click **Perform Check**.

Running Advisor on a Single Page

To run Advisor on a single page:

1. Navigate to the appropriate page.
2. Click the **Utilities** menu and select **Advisor**.
Advisor appears.
3. Under Checks to Perform, review the selected options. Select and deselect options as appropriate.
4. Click **Perform Check** at the top of the page.
A Results page appears.
5. To alter the existing preferences:
 - a. Under Filter Result, deselect the appropriate options.
 - b. Click **Apply Filter**
6. To re-run the Advisor and see if reported problems are fixed, click **Perform Check**.

Reviewing Session State

View current session state for your application by clicking the Session link on the Runtime Developer Toolbar.

Session state enables developers to store and retrieve values for a user as the user navigates between different application pages. Many applications are based on data contained within application controls. For example, buttons can display conditionally based on a value stored in session state.

See Also:

- ["Runtime Developer Toolbar"](#)
- ["Viewing Session State"](#)
- ["Managing Session State Values"](#)
- ["Managing Session State and User Preferences" in *Oracle Application Express Administration Guide*](#)

About Monitoring Application and Page Resource Use

Query the `V$SESSION` and `V$SQLAREA` views to monitor application and page resource use..

Oracle Application Express facilitates the monitoring of resources used by applications and pages by calling the package `DBMS_APPLICATION_INFO`. Whenever the Application Express engine renders or processes a page, the module is set to `APEX` and includes

the application ID and page number. Once set, you can query the `V$SESSION` and `V$SQLAREA` views to monitor transactions.

Enabling SQL Tracing and Using TKPROF

Enable SQL tracing and then analyze the temporary file Oracle Application Express creates using the TKPROF utility.

Tracing your session can be a very effective way to debug an application. From a database perspective, each page request is a single database session. If you enable SQL tracing, then Oracle Application Express creates a temporary file you can then analyze using the TKPROF utility.

You enable SQL tracing in Oracle Application Express by using `f?p` syntax to set the argument `p_trace=YES`. For example, to trace the display of page 1 in application 100, you would use the syntax:

```
http://.../f?p=100:1&p_trace=YES
```

To use the TKPROF utility:

1. Log in to SQL*Plus as a privileged user.
2. Execute the following statement:

```
show parameter USER_DUMP_DEST
```
3. Navigate to the directory in which the trace file is created.
4. Run the TKPROF utility from the operating system prompt using the following syntax:

```
tkprof filename1 filename2 [waits=yes|no] [sort=option] [print=n]  
[aggregate=yes|no] [insert=filename3] [sys=yes|no] [table=schema.table]  
[explain=user/password] [record=filename4] [width=n]
```

The input and output files are the only required arguments.

5. To view online Help, invoke TKPROF without arguments.



See Also:

"Tools for End-to-End Application Tracing" in *Oracle Database SQL Tuning Guide* for information about using the TKPROF program

Utilizing Debug Mode

Use Debug mode to track down unexpected application behavior.

- [About Debug Mode](#)
- [Enabling and Disabling Debug Mode for an Application](#)
- [Running an Application in Debug Mode](#)
- [Viewing Debug Reports in Development Mode](#)

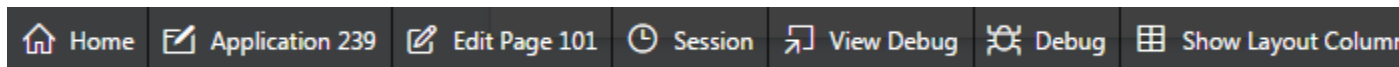
- [About Using f?p Syntax to Access Debug Mode](#)
- [Viewing Debug Reports from a Running Application](#)
- [Viewing Debug Reports](#)

 **See Also:**

"APEX_DEBUG" in *Oracle Application Express API Reference*

About Debug Mode

The Debug mode is a built in mechanism used to track down unexpected application behavior. You enable and disable debug mode for the current page and session by clicking the **Debug** and **No Debug** links in the Runtime Developer Toolbar. To view Debug reports, click the **View Debug** link in the Runtime Developer Toolbar.



If an application runs in debug mode, Oracle Application Express writes information about how it processes each page request to a log table. You can control the logging level of detail, add debug calls in your own code, and analyze the written log information.

When debug is enabled, Oracle Application Express also uses its non-minified JavaScript files to better isolate where possible issues may be coming from with client-side-code.

 **See Also:**

- ["Runtime Developer Toolbar"](#)
- ["APEX_DEBUG" in Oracle Application Express API Reference](#)

Enabling and Disabling Debug Mode for an Application

You can configure whether end users can run the application in debug mode by using the Debugging attribute on the Edit Application Definition page. A developer who is logged into workspace where the application resides can always run the application in debug mode.

To enable or disable debug mode:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Edit Application Properties** button to the right of the application name.

The Edit Application page appears.

4. Scroll down to Properties.
5. For Debugging:
 - **Yes.** Enables the application debug mode using the browser at runtime.
 - **No.** Disables the application debug mode using the browser at runtime.



See Also:

["Managing Application Attributes"](#)

Running an Application in Debug Mode

You run and disable debug mode at runtime by clicking **Debug** and **No Debug** on the Runtime Developer Toolbar. By default, Debugging mode is disabled, unless you are logged into the application's workspace as a developer.

To enable or disable Debug mode at runtime:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the desired application.
3. Run the application.
4. On the Runtime Developer Toolbar, click **Debug**.
The debugging mode is enabled. Debug data is captured in the background.
5. To disable debug mode, on the Runtime Developer Toolbar, click **No Debug**.



See Also:

["Enabling and Disabling Debug Mode for an Application"](#)

Viewing Debug Reports in Development Mode

To view debug reports in development mode:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the desired application.
3. Click **Utilities**.
4. On the Utilities page, click **Debug Messages**.
The Debug page appears.
5. Click the View Identifier to the left of the session you want to view.
The Debug Message Data page appears.

- To view steps that took the longest, hover over that step on the graph above the report.
The step details display.
- To go to that step in the report, click the bar.

 **Note:**

Debug sessions listed in the report live for at least 2 weeks and age out along with the activity log. This is usually more than adequate for debugging purposes.

 **See Also:**

["Enabling and Disabling Debug Mode for an Application"](#)

About Using f?p Syntax to Access Debug Mode

You can also use f?p syntax to run an application in Debugging mode. Simply call the page and set the Debug argument to YES or LEVEL n , where n is between 1 (least detailed) and 9 (most detailed). For example:

```
f?p=100:1:&APP_SESSION.::YES
```

 **See Also:**

["About Using f?p Syntax to Link Pages"](#)

Viewing Debug Reports from a Running Application

To view debug report from a running application:

- Locate and run the application in Debug mode.
- Click **View Debug** on the Runtime Developer Toolbar.
A list of Debug reports displays.
- Click the View Identifier to the left of the session you want to view.
The Debug Message Data page appears.
- To view steps that took the longest, hover over that step on the graph above the report.
The step details display.
- To go to that step in the report, click the bar.

 **Note:**

Debug sessions listed in the report live for at least 2 weeks and age out along with the activity log. This is usually more than adequate for debugging purposes.

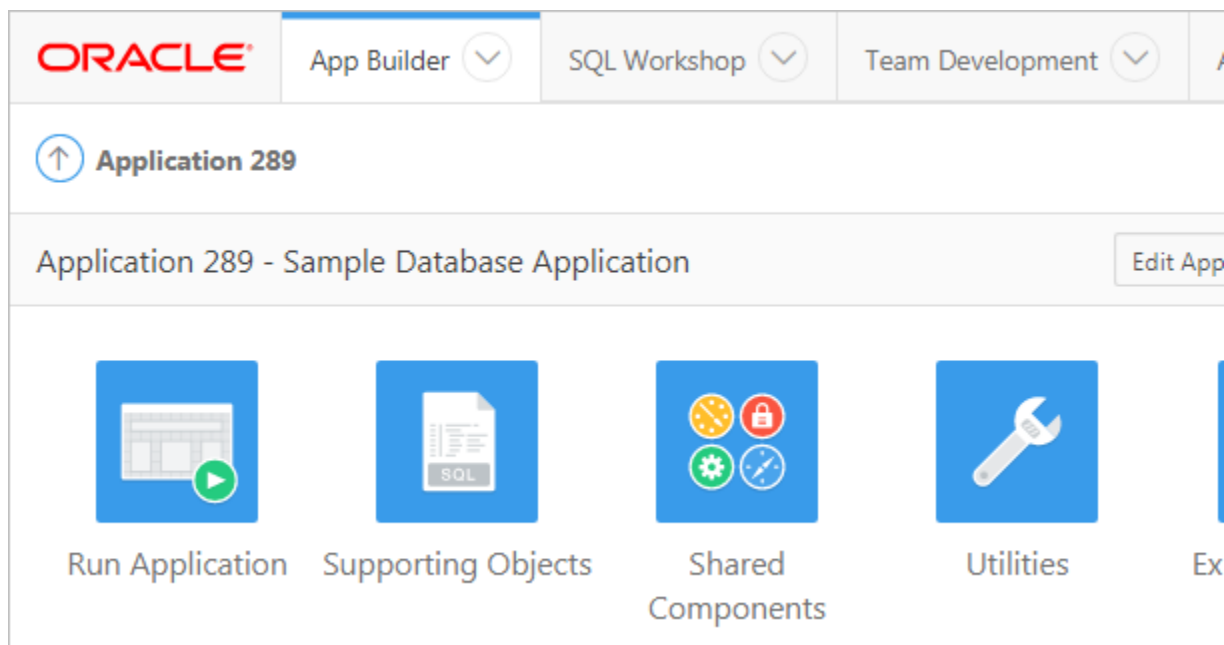
 **See Also:**

- ["Running an Application in Debug Mode"](#)
- ["Enabling and Disabling Debug Mode for an Application"](#)

Viewing Debug Reports

To view debug reports:

1. Click the **Find** icon.



The Items Finder appears.

2. Click the **Debug** tab.
3. Click the View Identifier to the left of the session you want to view.

The Debug report appears.

4. To view steps that took the longest, hover over that step on the graph above the report.

The step details display.

5. To go to that step in the report, click the bar.

 **Note:**

Debug sessions listed in the report live for at least 2 weeks and age out along with the activity log. This is usually more than adequate for debugging purposes.

 **See Also:**

["Enabling and Disabling Debug Mode for an Application"](#)

About Removing Controls and Components to Isolate a Problem

If you have problems running a page, try removing controls and components one at a time.

Using this approach, you can quickly determine which control or component may be the source of your problem. You can disable a control or component creating a build options. Build options enable you to conditionally display specific functionality within an application.

Build options enable you to conditionally display specific functionality within an application. Build options have two possible values: INCLUDE and EXCLUDE. If you specify an attribute as being included, then the Application Express engine considers it part of the application definition at runtime. Conversely, if you specify an attribute as being excluded, then the Application Express engine treats it as if it did not exist.

Disadvantages of Using Conditions to Remove Controls or Components

Prior releases of Oracle Application Express recommended that developers disable controls or components by setting the Condition attribute to **Never**. Although this approach is still valid, a major disadvantage is it overwrites the existing condition.

 **See Also:**

["Using Build Options to Control Configuration"](#)

Deploying an Application

Learn about best practices for deploying an application. Deployment is the process of moving an application from the testing phase to the production phase.

Tip:

To review recommendations about optimizing the development and deployment of Oracle Application Express applications throughout their life-cycle, see *Life Cycle Management with Oracle Application Express*. Go to <http://www.oracle.com/technetwork/developer-tools/apex/learnmore/apex-life-cycle-management-wp-3030229.pdf>.

Tip:

To review a tutorial that demonstrates deployment options, go to the Oracle Online Learning Library at <http://www.oracle.com/oll/apex> and search for *Advanced Deployment of Your Application*.

- [About Suggested Environments](#)
Oracle recommends developers follow standard system development life cycle practices when developing applications by having different environments for development, testing, and production.
- [System Development Life Cycle Methodologies to Consider](#)
The system development life cycle (SDLC) is the overall process of developing software using a series of defined steps. There are several system SDLC models that work well for developing applications in Oracle Application Express.
- [Understanding the Deployment Process](#)
Oracle recommends developing Oracle Application Express applications in separate environments: development, testing, and finally production. Learn about different approaches and best practices during when deploying applications.
- [How to Create a Custom Application](#)
Simplify the steps needed to deploy an application by creating a custom application using the Supporting Objects utility.
- [Using Build Options to Control Configuration](#)
Use build options to conditionally include or exclude application components and functionality.
- [About Publishing the Database Application URL](#)
Once you deploy your application, load the data, and create users, you can publish your production URL.

- [About Publishing the Websheet Application URL](#)
Once you complete your Websheet application, you can publish your production URL.
- [Exporting an Application and Application Components](#)
Developers can export an application and related files to another Oracle Application Express instance.
- [Importing Export Files](#)
Once you export an application and any related files, you must import them into the target Oracle Application Express instance and then install them.
- [Installing Export Files](#)
After you export an application and any related files and you import them into the target Oracle Application Express instance, the files are stored in the Export Repository. Next, you must install them.

 **See Also:**

- ["Oracle Application Express Administration Guide"](#)
- ["Extending Application Capabilities"](#)

About Suggested Environments

Oracle recommends developers follow standard system development life cycle practices when developing applications by having different environments for development, testing, and production.

As a best practice, developers should only make changes to applications and related database objects in the development environment. To further enforce this policy Oracle recommends that you install runtime-only Oracle Application Express environments in your test and production environments. This will prohibit developers from accessing the App Builder and SQL Workshop in these environments. As a general rule, your administrator (DBA) should be the only ones with permission to update the test and production environments. DBAs should use the appropriate provided APIs and import applications from an SQL interface, such as SQL Developer or SQL*Plus.

 **See Also:**

- ["About the Differences Between Runtime and Full Development Environments"](#)

System Development Life Cycle Methodologies to Consider

The system development life cycle (SDLC) is the overall process of developing software using a series of defined steps. There are several system SDLC models that work well for developing applications in Oracle Application Express.

- [About Iterative Vs Planned Development](#)
- [About the Advantages of Creating Prototypes](#)
- [Waterfall](#)
- [Spiral](#)
- [Rapid Application Development](#)

About Iterative Vs Planned Development

When developing applications using App Builder, you must find a balance between two dramatically different development methodologies:

- Iterative, rapid application development
- Planned, linear style development

Iterative, rapid application development offers so much flexibility that you run the risk of never completing your project. In contrast, Planned, linear style development can yield applications that do not meet the needs of end users even if they meet the stated requirements on paper.

About the Advantages of Creating Prototypes

The Oracle Application Express development environment enables developers to take a more iterative approach to development. Unlike many other development environments, creating prototypes is easy. With Oracle Application Express, developers can:

- Use built-in wizards to quickly design an application user interface.
- Make prototypes available to users and gather feedback.
- Implement changes in real time, creating new prototypes instantly.

Methodologies that work well with Oracle Application Express include Spiral and Rapid Application Development (RAD).

See Also:

- ["Spiral"](#)
- ["Rapid Application Development"](#)

Waterfall

The Waterfall is probably the best known SDLC model. In this methodology, the development process is broken down into the following stages:

1. Project Planning
2. Requirements Definition
3. Design
4. Development

5. Integration and Testing
6. Installation and Acceptance
7. Maintenance

This methodology is referred to as a waterfall because the output from one stage is the input for the next stage.

A primary problem with this approach is that it is assumed that all requirements can be established in advance. Unfortunately, requirements often change and evolve during the development process.

Spiral

A **Spiral** methodology is actually a series of short waterfall cycles. Each waterfall cycle yields new requirements and enables the development team to create a robust series of prototypes. One advantage of this approach is that it accommodates changing requirements. Disadvantages include complex project management and the risk development goes on indefinitely.

Rapid Application Development

A **Rapid Application Development (RAD)** methodology has a heavy emphasis on creating a prototype that closely resembles the final product. The prototype is an essential part of the requirements phase. Advantages of this model include the ability to accommodate changing requirements, rapid development cycles, and progress can be easily measured. The major disadvantage of this model is that the emphasis on prototyping can result in scope creep. As a result, developers can lose sight of their initial goals in the attempt to create the perfect application.

Understanding the Deployment Process

Oracle recommends developing Oracle Application Express applications in separate environments: development, testing, and finally production. Learn about different approaches and best practices during when deploying applications.

- [About Moving an Application to Another Instance](#)
- [Deployment Options to Consider](#)
- [How to Move an Application to Another Instance](#)

About Moving an Application to Another Instance

To move an application from one Oracle Application Express instance to another, you must move both the metadata and supporting objects used by the application as follows:

1. Move the application definition and all associated files.
2. Move the supporting objects. Review the Database Object Dependencies report to determine what objects to move.

 **See Also:**

- ["How to Move an Application to Another Instance"](#)
- ["Using the Database Object Dependencies Report"](#)
- ["How to Create a Custom Application"](#)

Deployment Options to Consider

When you develop an application, you create the application within a specific workspace. Each workspace has a unique ID and name. A common scenario is to create the application in a development instance and then deploy it to a production instance.

Deployment options to consider include:

1. **Use the same workspace and same schema.** Export and then import the application and install it using a different application ID. This approach works well when there are few changes to the underlying objects, but frequent changes to the application functionality.
2. **Use a different workspace and same schema.** Export and then import the application into a different workspace. This is an effective way to prevent a production application from being modified by developers.
3. **Use a different workspace and different schema.** Export and then import the application into a different workspace and install it so that it uses a different schema. This new schema needs to have the database objects required by your application.
4. **Use a different database with all its variations.** Export and then import the application into a different Oracle Application Express instance and install it using a different workspace, schema, and database.

Whether to Copy the Workspace

Deciding whether to copy an existing workspace is a matter of preference. Keep in mind that the production version must have access to all the appropriate objects. For example, you might want to copy a workspace in the following situations:

- When you want to keep the same application identifiers between development and other environments.
- When the application relies on Oracle Application Express authentication. Copying the workspace automatically migrates all the required user data.

Whether to Copy the Database

When deciding whether to copy the database, remember that the schema against which the application runs must have access to the same objects as the development instance. The actual name of the schema is unimportant. You can change it during the import process.

About the Application ID

It is not necessary to have matching application IDs for a development version and production version of an application. In fact, as a best practice never hard code the application ID into your application. Instead, use the application alias (defined on the Edit Application page), or use a built-in substitution string (such as `APP_ID` and `APP_ALIAS`). Using a substitution string is the better approach because it enables you to change the application ID without affecting any application functionality.

See Also:

- ["Using the Database Object Dependencies Report"](#)
- ["Name"](#) for information about defining an application alias
- ["Using Built-in Substitution Strings"](#) for information about using `APP_ID` and `APP_ALIAS`

How to Move an Application to Another Instance

Whether you are moving an application to another workspace or just making a copy of it, deployment involves the following steps:

1. Move the supporting database objects (if appropriate). Review the Database Object Dependencies report to determine what objects to move.
See ["Using the Database Object Dependencies Report."](#)
2. Package the application definition with its supporting objects to create a custom application.
See ["How to Create a Custom Application."](#)
3. Export the custom application.
See ["Exporting an Application and Application Components."](#)
4. Import the exported files into the target Oracle Application Express instance.
See ["Importing Export Files."](#)

Tip:

You can also move the application definition and all supporting objects manually. See ["Exporting an Application and Application Components."](#)

- [About Managing Database Objects](#)

About Managing Database Objects

Before you export an application and the appropriate related files, you must determine if you also need to migrate the database objects referenced by the application. If you

are unsure of which database objects to move, review the Database Object Dependencies report.

If the target schema is different from the schema used in the development environment, you must migrate the database objects referenced by the application. In many cases, this process can be as simple as using Oracle database export and import utilities to copy the application schema from the development environment to target instance. The following are two common scenarios where this approach does *not* work:

- When the object development schema refers to tablespaces to which the target instance schema does not have access.
- When the development instance schema has sample data that you do not want to migrate to the target instance schema.

If a database administrator or an Oracle Application Express administrator is the person responsible for exporting Oracle Application Express applications, be sure to clearly communicate if he or she:

- Should include all data when exporting your application.
- Should NOT include data from specific tables you identify.

 **Tip:**

"Using the Data Workshop to Manage Data" in *Oracle Application Express SQL Workshop Guide*

 **See Also:**

- ["Using the Database Object Dependencies Report"](#)
- ["How to Create a Custom Application"](#)

How to Create a Custom Application

Simplify the steps needed to deploy an application by creating a custom application using the Supporting Objects utility.

- [How Creating a Custom Application Simplifies Deployment](#)
- [Creating a Custom Application](#)
- [Adding an Access Control List to a Custom Application](#)
- [Installing Supporting Objects](#)
- [Deleting Supporting Objects Scripts, Messages, and Installation Options](#)
- [Upgrading a Custom Application](#)
- [Deinstalling Supporting Objects](#)

- [Viewing an Install Summary](#)



See Also:

- ["How to Move an Application to Another Instance"](#)
- ["Exporting an Application and Application Components"](#)

How Creating a Custom Application Simplifies Deployment

From a developer's perspective, importing and installing an application is a complicated process. First, you create the target database objects and seed data. Second, you import and install the application definition and all related files, including images, themes, and any other required static files.

Creating a custom application using the Supporting Objects utility greatly simplifies this process. Instead of performing numerous steps to create the database objects and then import and install the application and all supporting files, you can define the supporting objects so that the application and supporting files can be migrated in a few easy steps.

After users import and install the application definition, a wizard guides them through a few simple configuration steps. Then, the wizard asks whether to install the supporting application objects. Users have the option of installing the supporting application objects then or doing it later.

From a developer's perspective, creating a custom application has the following advantages:

- Ensures that the supporting objects are created in the correct order.
- Provides an automated process for deploying an application quickly using very few steps.
- Gives users the option to install supporting application objects when they import and install the application definition or at a later time.
- Enables users and developers with a convenient method for removing the application definition, supporting files, and all database objects.
- Provides users and developers with an easy way to upgrade a previously released custom application.

Plus, you can also take advantage of the Deinstall and Install features to quickly edit the underlying database objects that support an application. For example, you can deinstall and remove all database objects, edit the underlying database object creation scripts, and reinstall to create the redefined application objects.

 **See Also:**

- ["Installing Supporting Objects"](#)
- ["Deinstalling Supporting Objects"](#)
- ["Upgrading a Custom Application"](#)

Creating a Custom Application

To create a custom application, you must create installation scripts that define your application's supporting objects (including database objects, images, and seed data) and any preinstallation validations. You define these objects and the installation and deinstallation scripts and the messages that display when the user installs or deinstalls on the Supporting Objects page.

- [Accessing the Supporting Objects Utility](#)
- [Supporting Objects Page](#)

Accessing the Supporting Objects Utility

You create a custom application on the Supporting Objects utility.

To access the Supporting Objects Page utility:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
The Application home page appears.
3. Click the **Supporting Objects** icon.
The Supporting Objects page appears.

Supporting Objects Page

The top of the Supporting Objects page displays the application name and indicates current selections for the following: Check for Objects, Verify System Privileges, Required Free KB, Prompt for License, and Include in Export. To learn more about these options, see field-level Help.

To specify whether to include supporting objects with an application export, select *Yes* or *No* from the **Include in Export** list and click **Apply Changes**.

Application 252 \ Supporting Objects
?

Supporting Objects

Use this utility to define the database object definitions, images, and seed data to be included in your application export.

Application: **252: Sample Database Application** ?

Verify System Privileges: **Yes** ?

Prompt for License: **No** ?

Check for Objects: **No** ?

Required Free KB: **100** ?

Include in Export: **Yes** ?

| Installation | Upgrade | Deinstallation |
|--|---|--|
| <ul style="list-style-type: none"> Prerequisites Application Substitution ... 0 Build Options 0 Pre-installation Validatio... 0 Installation Scripts 4 Messages | <ul style="list-style-type: none"> Upgrade Scripts 5 Upgrade Message | <ul style="list-style-type: none"> Deinstallation Script 1 Deinstallation Message |

The rest of the page is divided into the following categories: Installation, Upgrade, and Deinstallation.

- [Installation](#)
- [Upgrade](#)
- [Deinstallation](#)

See Also:
["Viewing Field-Level Help"](#)

Installation

Use the links under Installation to define the following types of information:

- **Prerequisites.** Defines built-in checks required before installing the application, such as required free disk space, required system privileges, and schema object restrictions.

- **Application Substitution Strings.** Lists static substitution strings defined for the application. You can define static substitution strings for phrases or labels that occur in many places within an application.

When packaging an application, you can include prompts for substitution strings which users can specify when they install the custom application.
- **Build Options.** Lists build options defined for this application. You can use build options to conditionally display specific functionality within an application.

When packaging an application, you can include prompts for specific build options which display when the application is installed.
- **Pre-installation Validations.** Lists validations defined for the custom application. Similar to normal page validations, these validations prevent a user from installing database objects if the user-defined conditions are not satisfied. To create a new validation, click **Create** and follow the on-screen instructions.
- **Installation Scripts.** Enables a you to define multiple installation scripts that install supporting objects for the application. To create a new script, click **Create** and follow the on-screen instructions. To edit an existing script, click the **Edit** icon. To prevent another developer from editing a script, click the **Lock** icon. If the script is locked, the Lock icon appears as a closed padlock. If the script is unlocked, the Lock icon appears as an open padlock.
- **Messages.** Enables you to define messages that display when the user installs or deinstalls the application. Supported HTML tags include ``, `<i>`, `<u>`, `<p>`, `
`, `<hr>`, ``, ``, ``, and `<pre>`.

When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack.

See Also:

- ["Substitutions"](#)
- ["Using Build Options to Control Configuration"](#)
- ["Exporting Build Options or Build Option Status"](#)
- ["Understanding Cross-Site Scripting Protection"](#)

Upgrade

Use the links under Upgrade to define the following types of information:

- **Upgrade Scripts.** Click **Upgrade Scripts** to define scripts to upgrade database objects, images, and seed data when upgrading an existing application.
- **Upgrade Message.** Enables you to define messages that display when the user upgrades the application. Supported HTML tags include ``, `<i>`, `<u>`, `<p>`, `
`, `<hr>`, ``, ``, ``, and `<pre>`.

When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack.

 **See Also:**

- ["Upgrading a Custom Application"](#)
- ["Understanding Cross-Site Scripting Protection"](#)

Deinstallation

Use the links under Deinstallation to define the following types of information:

- **Deinstallation Script.** Click **Deinstallation Script** to define a script to drop database objects and static files created by the installation scripts. To edit an existing script, click the **Edit** icon.
- **Deinstallation Message.** Allows you to define messages that display when the user deinstalls the application. Supported HTML tags include ``, `<i>`, `<u>`, `<p>`, `
`, `<hr>`, ``, ``, ``, and `<pre>`.

When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack.

 **See Also:**

- ["Understanding Cross-Site Scripting Protection"](#)

Adding an Access Control List to a Custom Application

You can control access to an application, individual pages, or page components by creating an access control list.

To add an access control list of a custom application:

1. Create an access control list.
2. Navigate to the Supporting Objects page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select the application.
The Application home page appears.
 - c. Click **Supporting Objects**.
The Supporting Objects page appears.
3. Under Installation, click **Installation Scripts**.
The Installation Scripts page appears.
4. Click **Create**.
5. At the bottom of the page, expand the **Tasks** region and click **Create Scripts for Access Control Tables**.

If Access Control tables are defined, the Create Script page displays the tables to be included.

6. Click **Create Script**.



See Also:

["Controlling Access to Applications, Pages, and Page Components"](#)

Installing Supporting Objects

After you edit your supporting objects and create the appropriate scripts, you can run your installation scripts by clicking **Install Supporting Objects** on the Tasks list.

End users can also use this feature if they elect to not install the custom application (or supporting objects) after they import and install the application definition.

To install supporting objects:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Supporting Objects**.

The Supporting Objects page appears.

4. Click the **Install Supporting Objects** on the Tasks list.
5. To view details about the installation script before running it, expand the Tasks region and click **Preview Installation Script**.

The Preview Scripts page appears listing summary information, prerequisites, and the actual scripts to be run.

6. To exit the Preview Scripts page and continue, click **Close**.
7. From Install Supporting Objects, click **Yes** and click **Next**.
8. Follow the on-screen instructions.

Deleting Supporting Objects Scripts, Messages, and Installation Options

You can delete the metadata that defines supporting object scripts, messages, and installation options associated with a custom application by clicking **Remove Supporting Object Installation** on the Tasks list on the Supporting Objects page.

To delete the metadata that defines supporting object scripts, messages, and installation options:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Supporting Objects**.

The Supporting Objects page appears.

4. On the Tasks list on the right side of the page, click **Remove Supporting Object Installation**.
5. Follow the on-screen instructions.

Upgrading a Custom Application

You can define scripts to upgrade a previously published application on the Upgrade page.

- [Defining an Upgrade Script](#)
- [Upgrading a Custom Application](#)

Defining an Upgrade Script

You can use the Upgrade page to define scripts to upgrade database objects, images, and seed data when upgrading an existing application.

To create an upgrade script:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Supporting Objects**.
The Supporting Objects page appears.
4. Under Upgrade, click **Upgrade Scripts**.
5. Use the Detect Existing Supporting Objects section to determine if the appropriate objects are installed or must be upgraded.
6. In Query to Detect Existing Supporting Objects, enter a query in the field that returns at least one row if the supporting objects exist.
This query determines whether the user who installs the custom application is prompted to run the installation scripts or the upgrade scripts.
7. To create a script, click **Create**.

Tip:

To enable users to upgrade from various earlier versions of this application, you can add conditions to the upgrade scripts by going to the Script Properties page.

8. To prevent another developer from editing a script, click the **Lock** icon.
If the script is locked, the Lock icon appears as a closed padlock. If the script is unlocked, the Lock icon appears as an open padlock.
9. To edit an existing script, click the **Edit** icon.
10. Follow the on-screen instructions.

Upgrading a Custom Application

After you create your upgrade script, you can test it by clicking **Upgrade Supporting Objects** on Tasks list on the Supporting Object page.

End users can also use this feature to upgrade an existing custom application.

To upgrade a custom application:

1. Import a new version of application to be upgraded (if applicable).
2. On the Workspace home page, click the **App Builder** icon.
3. Select the application.
4. Click **Supporting Objects**.
The Supporting Objects page appears.
5. From the Tasks list, click **Upgrade Supporting Objects**.
6. Follow the on-screen instructions.



See Also:

["Importing an Application, Page or Component Export"](#)

Deinstalling Supporting Objects

Once you create or install a custom application, you can deinstall it by either:

- Clicking the **Deinstall Supporting Objects** on the Supporting Objects page.
- Clicking **Delete this Application** on the Application home page.

When you deinstall an application, you have the option of removing the current application definition and running the deinstallation script defined in the Supporting Objects.

To deinstall a custom application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Supporting Objects**.
The Supporting Objects page appears.
4. From the Tasks list, click **Deinstall Supporting Objects**.
5. Select a deinstallation option:
 - **Remove Application Definition** removes the current application definition.
 - **Deinstall Database Objects** runs the deinstallation script defined in the deployment attributes for this application.
6. Follow the on-screen instructions.



See Also:

"Creating a Custom Application"

Viewing an Install Summary

You can view a log of recent installation and deinstallation by clicking **View Install Summary** on the Tasks list on the Supporting Objects page. Note that this log only displays results from the most recent installation or deinstallation that occurred during the current Application Express session.

To view the Install Summary:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Supporting Objects**.
The Supporting Objects page appears.
4. On the Tasks list on the right side of the page, click **View Install Summary**.
A Summary page appears.

Using Build Options to Control Configuration

Use build options to conditionally include or exclude application components and functionality.

- [About Build Options](#)
Build options enable developers to enable or disable application components and functionality when the application installs or at runtime using the `APEX_UTIL.GET_BUILD_OPTION_STATUS` and `APEX_UTIL.SET_BUILD_OPTION_STATUS` APIs.
- [Creating Build Options](#)
Create build options by selecting the application and accessing Build Options on the Shared Components page.
- [Including or Excluding Build Options](#)
Specify **Include** to enable a component and include it with the application. Specify **Exclude** to disable a component and exclude it from the application.
- [Selecting a Build Option](#)
Apply build options to a page, component, page control, or shared component in Page Designer.
- [Deleting Build Options and Associated Components](#)
Delete a build option by first removing the associated components and then deleting the build option.
- [Viewing Build Option Reports](#)
Access the Utilization report where build options are utilized in the current application. View the History report to see what modifications have been made to build options in the current application.

- [Exporting Build Options or Build Option Status](#)
Export build option status to toggle build options on or off within another environment.

About Build Options

Build options enable developers to enable or disable application components and functionality when the application installs or at runtime using the `APEX_UTIL.GET_BUILD_OPTION_STATUS` and `APEX_UTIL.SET_BUILD_OPTION_STATUS` APIs.

You can apply build options to most application components (such as pages, regions, items, validations, and so on) and specify whether to include or exclude them in the runtime application.

Build options have two possible values:

- **Include** - Application components are enabled and included with the application.
- **Exclude** - Application components are disabled and excluded from the application.

If you specify **Include**, then the Application Express engine considers the component as part of the application definition at runtime. Conversely, if you specify **Exclude**, then the Application Express engine treats the component as if it did not exist.

See Also:

`APEX_UTIL.GET_BUILD_OPTION_STATUS` and
`APEX_UTIL.SET_BUILD_OPTION_STATUS` in *Oracle Application Express API Reference*

Use Case 1: Enable and Disable Functionality Based on Installation Type

You are developing an application that has many installation types. For each installation type, users may choose to enable or disable certain application functionality. With build options, you can develop one application and when it installs, you can configure build options to include or exclude associated functionality. You may also use an API to expose features within the application. This enables privileged end users to selectively enable or disable application functionality.

Use Case 2: Hiding Incomplete Functionality in Various Environments

You are developing functionality which is not yet complete, so you do not wish to expose it to end users. You can use build options to disable functionality when the application is built for test and production environments. The benefit of this approach is that the application can still be deployed without including incomplete functionality.

Use Case 3: Determining the Consequences of Removing Functionality

You are removing functionality you think is not needed by the application, but are unsure if removing the functionality is a safe operation. In this case, you can use build options to essentially "comment out" specific functionality. When the application is deployed and you determine the functionality is necessary, you can re-instate it. If the

functionality is deemed as unnecessary, you can then safely remove it from the application.

Creating Build Options

Create build options by selecting the application and accessing Build Options on the Shared Components page.

To create a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
5. To create a new build option, click **Create**.
6. Under Attributes:
 - a. Application - Identifies the application to which this build option applies.
 - b. Build Option - Enter a descriptive name for this build option. Build options are predefined settings that determine whether or not components within an application are enabled.
 - c. Status - Options:
 - **Include** - Include this feature or component. Associated application components are enabled and included with the application.
 - **Exclude** - Do not include this feature or component. Associated application components are disabled and excluded from the application.
 - d. Default on Export - When this application is exported, set the build option to this value.
 - e. On Upgrade Keep Status - Select whether the build option status in the deployed application should be kept, or overwritten with the specified status when the application is upgraded. **Yes** keeps the deployed status and is useful if the application supports build option configuration (using the `APEX_UTIL.SET_BUILD_OPTION_STATUS` API) to enable or disable the build option in the deployed application, and where you want to respect those settings upon upgrade.
 - f. Comments - Enter any comments or notes. These comments never display when running the application.
7. Click **Create Build Option**.

Including or Excluding Build Options

Specify **Include** to enable a component and include it with the application. Specify **Exclude** to disable a component and exclude it from the application.

Build options have two possible values: **Include** and **Exclude**. If you specify an attribute as being included, then the Application Express engine considers it part of the application definition at runtime. Conversely, if you specify an attribute as being excluded, then the Application Express engine treats it as if it did not exist.

To include or exclude a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
The Build Options page appears.
5. Select a build option name.
The Create/Edit Build Option page appears.
- 6.
7. For Status, select either:
 - **Include** - Include this feature or component. Associated application components are enabled and included with the application.
 - **Exclude** - Do not include this feature or component. Associated application components are disabled and excluded from the application.
8. Edit any additional attributes as needed.
To learn more about an attribute, see field-level Help.
9. To save your changes, click **Apply Changes**.

 **See Also:**

`APEX_UTIL.GET_BUILD_OPTION_STATUS` and
`APEX_UTIL.SET_BUILD_OPTION_STATUS` in *Oracle Application Express API Reference*

Selecting a Build Option

Apply build options to a page, component, page control, or shared component in Page Designer.

Once you create a build option, you can select it for a page, a component (report, chart, or form), a specific page control (button, item, list of value), and another shared component (breadcrumb, list, or tab). You apply build options to a page, component, page control, or shared component by navigating to the appropriate page. In Page Designer, select the page, component, control, or shared component and find **Configuration, Build Option**.

Deleting Build Options and Associated Components

Delete a build option by first removing the associated components and then deleting the build option.

- [Removing Components Associated with a Build Options](#)
- [Deleting Build Options Permanently](#)

Removing Components Associated with a Build Options

Removing components associated with a build option permanently deletes them.

To remove components associated with a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
The Build Options page appears.
5. Select the build option .
The Create/Edit Build Option page appears. Note that Associated Components display at the bottom of the page.
6. Scroll down to Associated Components and click **Remove Components**.
A verification window appears.
7. Click **Yes - remove components** and then click **Remove Components**.
The components are removed.

Deleting Build Options Permanently

Tip:

Deleting a build option does not delete the Associated Components. Before deleting a build option, review the Associated Components to determine if the components should be removed.

To delete a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
The Build Options page appears.
5. Select the build option.
The Create/Edit Build Option page appears.
6. Click **Delete**.
A verification window appears.
7. Click **Delete Build Option**.
The build option is removed.

Viewing Build Option Reports

Access the Utilization report where build options are utilized in the current application. View the History report to see what modifications have been made to build options in the current application.

- [Viewing Build Option Utilization](#)
- [Viewing Build Option History](#)

Viewing Build Option Utilization

Once you create a build option, a Utilization tab appears on the Build Options page. This report details build option utilization in the current application.

 **Note:**

The Utilization tab only appears on the Build Options page after you create a build option.

To view the Build Option Utilization report:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
5. On the Build Options page, click **Utilization**.

The Build Option Utilization report appears.

6. Select a build option and click **Go**.

Viewing Build Option History

To view the Build Option History report:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
5. On the Build Options page, click **History**.

The Build Option History report appears.

Exporting Build Options or Build Option Status

Export build option status to toggle build options on or off within another environment.

You can export build options or build option status on the Component Export page. For example, you can use this feature to deploy a production application with a hidden feature by associating the components of the hidden feature with a build option having the status of `EXCLUDE`. After deployment, you can enable the hidden feature by changing the status of the build option to `INCLUDE` and then exporting the Build Option Status.

Once you apply the Build Options Status to the production instance, the new feature appears.

To export build options or build option status:

1. Navigate to the Component Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. From the Tasks list, click **Export Application Components**.
2. On the Component Export page, select the build options to export:
 - a. Click the **Build Options Status** tab and select the build options to export.
 - b. Click **Add to Export**.
3. Click **Next**.
4. On Components Export:
 - a. **File Format** - Select how rows in the export file are formatted:
 - **UNIX** - The resulting file contains rows delimited by line feeds.
 - **DOS** - The resulting file contains rows delimited by carriage returns and line feeds.
 - b. **As of** - Specify a time in minutes to go back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

This utility uses the `DBMS_FLASHBACK` package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo tablespace. While two databases may have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo tablespace, forcing older data to be archived.

- c. Click **Export Components**.

 **Tip:**

For translated applications, the Export Translations field displays. To export the selected shared components for the primary language application and all translated applications, from Export Translations field, select **Yes**. To ensure text strings and markup are up-to-date, remember to synchronize the translated applications prior to exporting the translated components.

 **See Also:**

- ["How to Create a Custom Application"](#)
- ["Exporting Application Components"](#)

About Publishing the Database Application URL

Once you deploy your application, load the data, and create users, you can publish your production URL.

You can determine the production URL for your application by either:

- Selecting the application on the Application home page and right-clicking the **Run** button. Then, select **Copy link address** or **Copy link location** depending on your browser.
- Running the application and then copying the URL.

The Run button gets its value from the Home link attribute on the Edit Security Attributes page. This link is only referenced by this icon and by applications that do not use the Oracle Application Express Login API. Consider the following example:

```
http://apex.somewhere.com/pls/apex/f?p=11563:1:3397731373043366363
```

Where:

- `apex.somewhere.com` is the URL of the server.
- `pls` is the indicator to use the `mod_plsql` cartridge.
- `apex` is the database access descriptor (DAD) name. The DAD describes how Oracle HTTP Server connects to the database server so that it can fulfill an HTTP request. The default value is `apex`.
- `f?p=` is a prefix used by Oracle Application Express.
- `11563` is the application being called.
- `1` is the page within the application to be displayed.
- `3397731373043366363` is the session number.

To run this example, a user would use the URL:

```
http://apex.somewhere.com/pls/apex/f?p=11563:1
```

 **Tip:**

When users log in, they receive unique session numbers. This number displays after `:home` in the URL. Do not include the session number as part of the URL. When another user tries to run a URL containing a session ID, an error displays.

 **See Also:**

["Accessing the Edit Application Definition Page"](#)

About Publishing the Websheet Application URL

Once you complete your Websheet application, you can publish your production URL.

You can determine the URL to your Websheet application by running the application and copying the URL.

Consider the following example:

```
http://apex.somewhere.com/pls/apex/ws?p=123:home
```

Where:

- `apex.somewhere.com` is the URL of the server.
- `pls` is the indicator to use the `mod_plsql` cartridge.
- `apex` is the database access descriptor (DAD) name. The DAD describes how Oracle HTTP Server connects to the database server so that it can fulfill an HTTP request. The default value is `apex`.
- `ws?p=` is a prefix used by Oracle Application Express to link to a Websheet application.
- `123` is the Websheet application being called.
- `home` is the home page alias defined as Home Page on the Application Properties page.

To run this example application, you would use the URL:

```
http://apex.somewhere.com/pls/apex/ws?p=123:home
```

 **Tip:**

When users log in, they receive unique session numbers. This number displays after `:home` in the URL. Do not include the session number as part of the URL. When another user tries to run a URL containing a session ID, an error displays.

Exporting an Application and Application Components

Developers can export an application and related files to another Oracle Application Express instance.

Tip:

You can also export workspaces, applications, and related files from a command-line using the `APEXExport` program located in the `Utilities` folder in the downloaded installation zip file. See “Exporting Applications and Workspaces from a Command-line” in *Oracle Application Express Administration Guide*

- [About the Export Process](#)
- [About The Export Page](#)
- [Exporting an Application](#)
- [Exporting Application Components](#)
- [Exporting a Workspace](#)
- [Exporting a Page in an Application](#)
- [About Exporting Static Files](#)
- [Exporting Themes](#)
- [Exporting Plug-ins](#)
- [Exporting User Interface Defaults](#)
- [Exporting Team Development Feedback](#)
- [Exporting Script Files from the Script Repository](#)

About the Export Process

Exporting an application from Oracle Application Express is very straightforward process and produces a readable script file with a `.SQL` extension. You can run this SQL script in any Oracle Application Express environment which is the same release or later than the environment from which you are exporting. For example, an application exported from Oracle Application Express release 4.0 can be imported into an environment running Oracle Application Express release 4.0, 4.1, or 4.2 or a later release. However, you cannot export an application to an earlier Oracle Application Express release (for example, an application exported from Application Express release 4.2 cannot be imported into an environment running Application Express 4.1 or earlier).

An application export includes the application definition, supporting objects, and shared components (including plug-ins, images, CSS files, JavaScript files and other files which must be managed independently). Instead of exporting complete applications you can also choose to export specific components, such as a page. This also creates a SQL script file. There are limitations when importing components into a different environments. As a best practice, Oracle recommends you export complete

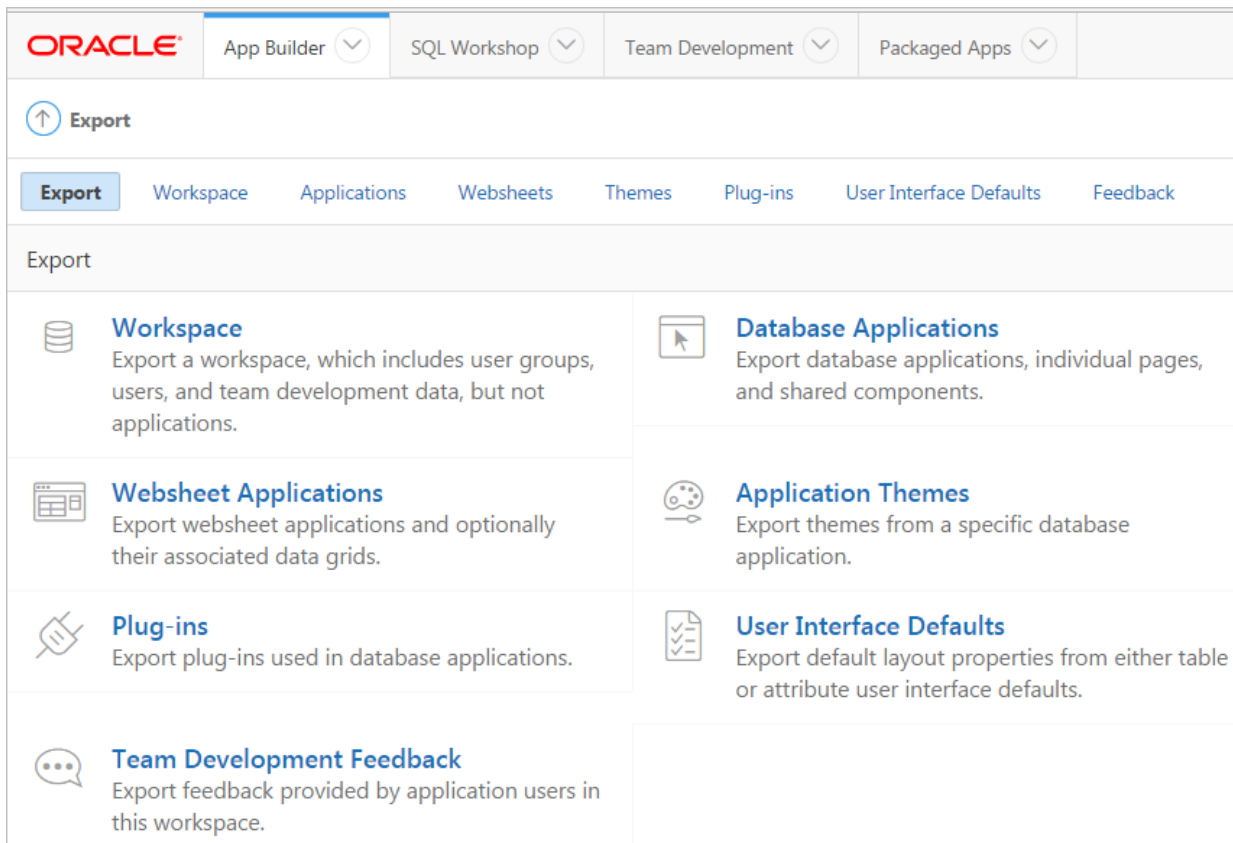
applications rather than individual components. If you wish to export an application and there are components, such as pages that are not yet ready for testing, Oracle recommends you use Build Options to include or exclude various application components.

 **See Also:**

- ["Exporting an Application"](#)
- ["Exporting Application Components"](#)
- ["Using Build Options to Control Configuration"](#)

About The Export Page

You export an application definition and all associated files using the following tabs at the top of the Export page: Export, Workspace, Applications, Websheet, Themes, Plug-ins, User Interface Defaults, and Feedback.



Workspace
Export a workspace, which includes user groups, users, and team development data, but not applications.

Database Applications
Export database applications, individual pages, and shared components.

Websheet Applications
Export websheet applications and optionally their associated data grids.

Application Themes
Export themes from a specific database application.

Plug-ins
Export plug-ins used in database applications.

User Interface Defaults
Export default layout properties from either table or attribute user interface defaults.

Team Development Feedback
Export feedback provided by application users in this workspace.

You do not need to export an entire workspace unless you want to migrate workspace users or replicate shared component subscriptions in the target instance.

Once you export an application and any related files, you must import them into the target Oracle Application Express instance and then install them. As a general rule, always import the application first and then the related files.

 **Tip:**

You can simplify the steps needed to deploy an application by creating a custom application. See:

- ["How to Create a Custom Application"](#)
- ["How to Move an Application to Another Instance"](#)

Exporting an Application

When you export an application, Oracle Application Express generates a text file containing PL/SQL API calls. An application export includes the application definition, supporting objects, and shared components (including plug-ins, images, CSS files, JavaScript files, and other files which must be managed independently).

To export an application:

1. Navigate to the Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Workspace Utilities**.
 - c. Click **Export**.
2. Click the **Applications** tab.
3. Under Choose Application, select the application to export.
4. Under Export Application:
 - a. **File Format** - Select how rows in the export file are formatted. Options:
 - **UNIX** - The resulting file contains lines terminated by line feeds.
 - **DOS** - The resulting file contains lines terminated by carriage returns and line feeds.
 - **Database** - Saves the resulting file to the Export Repository instead of downloading it.
 - b. **Owner Override** - By default, the application is owned by the owner identified in the application attribute owner. Use this attribute if you wish to import this application as a different schema owner than you are exporting.
 - c. **Build Status Override** - Select the build status of the exported application. Options include:
 - **Run Application Only** - Users can only run an application. Selecting this option is an effective way to protect an application from modifications from other developers.
 - **Run and Build Application** - Developers and users can both run and edit an application.

 **Tip:**

If you select **Run Application Only**, you cannot set the argument `p_trace` to `Yes`. The only way to change this setting after you import the application, is to log in to Oracle Application Express Administration Services. .

d. Debugging - Options:

- **Yes** - Exports the application with debugging enabled.
- **No** - Exports the application with debugging disabled.

e. As of - Specify a time in minutes to go to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

This utility uses the `DBMS_FLASHBACK` package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo tablespace. While two databases can have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo tablespace, forcing older data to be archived.

5. Under Export Preferences:**a. Export Supporting Object Definitions** - Specify whether to export supporting object definitions with your application.

- **Yes** - Includes supporting object definitions in the application export. Does not automatically load supporting objects when invoked from a command line.
- **No** - Does not include the supporting object definitions in the export.
- **Yes and Install on Import Automatically** - Includes supporting object definitions and a call to install supporting objects in the application export. This option is valid only for command line installs. When application imports from the command line, it automatically installs or upgrades the supporting objects.

Supporting object definitions include all configuration options and scripts and enable an application export to include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file.

b. Export Public Reports - Specify whether to include public reports with your application. Public reports are the public shared report settings saved by users from an interactive report or interactive grid.

- **Yes** - Includes Public reports in the application export.
- **No** - Does not include Public reports in the application export.

c. Export Private Reports - Specify whether to include Private reports with your application. Private reports are the private customized settings saved by users from an interactive report or interactive grid.

- **Yes** - Includes public reports in the application export.

- **No** - Does not include public reports in the application export.
- d. **Export Report Subscriptions** - Specify whether to include interactive report or interactive grid subscription settings with your application. This will be limited to the unexpired subscription settings of all users for all interactive reports or interactive grids in the application.
- **Yes** - Includes report subscriptions for the reports you are exporting in the application export.
 - **No** - Does not include report subscriptions for the reports in the application export.
- e. **Export Developer Comments** - Specify whether to include Developer Comments in your application export.
- **Yes** - Includes Developer Comments in the application export.
 - **No** - Does not include Developer Comments in the export.
- f. **Export Translations** - Specify whether to include Translations with your application export.
- **Yes** - Includes the translation mappings and all text from the translation repository.
 - **No** - Does not include translation mappings or the translation repository.
- Translation Text Messages and Dynamic Translations are always included in the application export.
- g. **Export with Original IDs** - Specify whether the export file should contain the application component IDs as of now or as of the last import of this application.
- **Yes** - Includes the IDs as of the last import.
 - **No** - Includes the current IDs of the components.
6. Click **Export**.

 **Tip:**

In addition to exporting the actual application file, you may also need to export other related files such as cascading style sheets, images, and script files.

 **See Also:**

- ["Using Build Options to Control Configuration"](#)
- ["Changing Application Build Status Set During Deployment"](#) in *Oracle Application Express Administration Guide*
- ["Importing Export Files"](#)
- ["Installing Exported Applications into a Runtime Environment"](#) in *Oracle Application Express Administration Guide*

Exporting Application Components

You can use the Component Export Wizard to export shared components or page components to another application or workspace, back up a component before editing it, or create an export that functions as a patch to another Oracle Application Express instance.

To export shared components or page components:

1. Navigate to the Component Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Click **Shared Components**.
 - d. From the Tasks list, click **Export Application Components**.
The Component Export page appears.
2. Click the following tabs and select the components to export:
 - **Components** displays shared application components and entire pages. Use the navigation bar at the top of the page to search for components.
 - **Components by Page** lists components of the selected page. Navigate to a specific page by making a selection from the Page list.
 - **Build Option Status** displays available build options. Use this page to turn build options on and off.

For each tab, select the appropriate components and click **Add to Export**.

3. Click **Next**.
4. For Component Export:
 - a. **File Format** - Select how rows in the export file are formatted:
 - **UNIX** - The resulting file contains lines terminated by line feeds.
 - **DOS** - The resulting file contains lines terminated by carriage returns and line feeds.
 - b. **As of** - Specify a time in minutes to go back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

This utility uses the `DBMS_FLASHBACK` package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo tablespace. While two databases may have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo tablespace, forcing older data to be archived.

- c. Click **Export Components**.

 **Tip:**

For translated applications, the Export Translations field displays. To export the selected shared components for the primary language application and all translated applications, from Export Translations field, select **Yes**. To ensure text strings and markup are up-to-date, remember to synchronize the translated applications prior to exporting the translated components.

 **See Also:**

- ["Managing Shared Components"](#)
- ["Importing an Application, Page or Component Export"](#)
- ["Exporting Build Options or Build Option Status"](#)

Exporting a Workspace

You can move or copy a workspace to another Oracle Application Express instance by exporting a workspace. When you export a workspace, Oracle Application Express creates an ASCII text SQL script of users, defined user groups, Team Development data, and workspace artifacts including SQL Scripts, SQL Command History, saved SQL, user preferences, developer login history, email logs, and user interface defaults.

To export a workspace:

1. Navigate to the Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Workspace Utilities**.
 - c. Click **Export**.
2. On the Export page, click the **Workspace** tab.
3. Under Export Workspace:
 - a. **Include Team Development** - Select **Yes** to include Team Development data such as Features, Milestones, To Do, Bugs, Feedback, Links and News. Only select **Yes** if the target instance for this workspace is not a runtime only instance. Otherwise, select **No**.
 - b. **Export Type** - In most cases, you should not change the default value **Minimal**. Only select **Full** if you want to replicate to another instance all workspace artifacts including SQL Scripts, SQL Command History, saved SQL, user preferences, developer login history, email logs, and user interface defaults.
 - c. **File Format** - Select how rows in the export file are formatted:
 - **DOS** - The resulting file contains lines terminated by line feeds.
 - **UNIX** - The resulting file contains lines terminated by carriage returns and line feeds.

- d. **File Character Set** - The export file will be encoded in the specified character set.
4. Click **Export**.



See Also:

"Managing Application Express Users" in *Oracle Application Express Administration Guide*

Exporting a Page in an Application

 **WARNING:**

Exporting and importing a single page to a different system is an advanced deployment technique with the following restrictions:

- Both systems have to have the same application ID and workspace ID.
- Both systems have to have the same IDs for Shared Components.
- If the page references new or updated Shared Components, those have to be exported and imported first.

If these restrictions are not fulfilled, the import will fail.

To export a page in an application:

1. Navigate to the appropriate application page:
2. Click the **Utilities** menu and select **Export**.
The Export Page Wizard appears.
3. For Page, identify the page you wish to export. You can select only one page for export at a time.
4. From File Format, select how rows in the export file are formatted. Options include:
 - **UNIX** - The resulting file contains rows delimited by line feeds.
 - **DOS** - The resulting file contains rows delimited by carriage returns and line feeds.
5. For **As of**, specify a time in minutes to go back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

This utility uses the `DBMS_FLASHBACK` package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo tablespace. While two databases may have the

same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo tablespace, forcing older data to be archived.

6. Click **Export Page**.

- [Exporting a Websheet](#)

 **See Also:**

- ["Viewing a Page in Page Designer"](#)
- ["Exporting Application Components"](#)
- ["Exporting an Application"](#)
- ["Importing an Application, Page or Component Export"](#)

Exporting a Websheet

Use the Export Websheet utility to make a Websheet application available to other users.

To export a Websheet application:

1. Navigate to the Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Workspace Utilities**.
 - c. Click **Export**.
2. On the Export page, click the **Websheets** tab.
3. On Export Websheet:
 - a. **Websheet Application** - Select the Websheet to export.
 - b. **File Format** - Select how rows in the export file are formatted. Options:
 - **UNIX** - The resulting file contains lines terminated by line feeds.
 - **DOS** - The resulting file contains lines terminated by carriage returns and line feeds.
 - c. **Export Preferences** - Select the appropriate options.
 - d. Click **Export**.

 **See Also:**

- ["Creating Websheet Applications"](#)
- ["Importing a Websheet"](#)

About Exporting Static Files

You can export static application files and static workspace files by creating a ZIP file.



See Also:

- ["Downloading All Static Application Files in a Zip"](#)
- ["Downloading All Static Workspace Files in a Zip"](#)

Exporting Themes

Use the Export Theme utility to export themes from one Oracle Application Express development instance to a file.

To export an application theme:

1. Navigate to the Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Workspace Utilities**.
 - c. Click **Export**.
2. Click the **Themes** tab.
3. On the Themes page:
 - a. Application - Select an application.
 - b. **Export Theme** - Select the theme to export.
 - c. **File Format** - Select how rows in the export file are formatted. Options include:
 - **UNIX** - The resulting file contains lines terminated by line feeds.
 - **DOS** - The resulting file contains lines terminated by carriage returns and line feeds.
 - d. Click **Export**.



See Also:

- ["Importing Themes"](#)
- ["Using Themes"](#)

Exporting Plug-ins

Use the Export Plug-in utility to export plug-ins from one Oracle Application Express development instance to a file.

To export a Plug-in from the Export page:

1. Navigate to the Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Workspace Utilities**.
 - c. Click **Export**.
2. Click the **Plug-ins** tab.
3. On the Plug-ins page:
 - a. **Application** - Select an application.
 - b. **Plug-in** - Select the plug-in to export.
 - c. **File Format** - Select how rows in the export file are formatted. Options include:
 - **UNIX** - The resulting file contains rows delimited by line feeds.
 - **DOS** - The resulting file contains rows delimited by carriage returns and line feeds.
 - d. Click **Export**.



See Also:

["Importing Plug-ins"](#)

Exporting User Interface Defaults

This section describes how to export User Interface Defaults.

- [About Exporting User Interface Defaults](#)
- [Exporting User Interface Defaults from the Export Page](#)
- [Exporting User Interface Defaults from the User Interface Defaults Page](#)

About Exporting User Interface Defaults

Exporting User Interface Defaults is useful when you plan to develop on a target system. User Interface Defaults are divided into two categories, the Table Dictionary, and the Attribute Dictionary.

- The Table Dictionary consists is specific to tables and columns within a selected schema. These defaults are used over those defined in the Attribute Dictionary.
- The Attribute Dictionary contains a set of attributes about a column that is used in creating forms and reports. The definitions are matched by column name and a particular definition can be shared among several columns by using synonyms.

When you export User Interface Defaults, all User Interface Defaults for the selected dictionary and schema are exported to a single SQL Command script. When prompted, save this file to your hard drive. The file contains an API call to create table hints by making calls to the application PL/SQL API. You can use this file to import User Interface Defaults to another database and Oracle Application Express instance.



See Also:

- ["Importing User Interface Defaults"](#)
- ["Managing User Interface Defaults"](#) in *Oracle Application Express SQL Workshop Guide*

Exporting User Interface Defaults from the Export Page

To export User Interface Defaults from the Export page:

1. Navigate to the Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Workspace Utilities**.
 - c. Click **Export**.
2. Click the **User Interface Defaults** tab.
3. On the Table Dictionary page:
 - a. **Schema** - Select the schema that owns the table associated with the User Interface Defaults.
 - b. **File Format** - Select how rows in the export file are formatted. Options include:
 - **UNIX** - The resulting file contains lines terminated by line feeds.
 - **DOS** - The resulting file contains lines terminated by carriage returns and line feeds.
 - c. Click **Export**.
4. On the Attribute Dictionary page:
 - a. **File Format** - Select how rows in the export file are formatted. Options include:
 - **UNIX** - The resulting file contains rows delimited by line feeds.
 - **DOS** - The resulting file contains rows delimited by carriage returns and line feeds.
 - b. Click **Export**.

Exporting User Interface Defaults from the User Interface Defaults Page

To export User Interface Defaults from the User Interface Defaults page:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **User Interface Defaults**.
The User Interface Defaults page appears.
4. Select a dictionary:

- **Table Dictionary** - Exports specific tables and columns within a selected schema.
 - **Attribute Dictionary** - Exports a set of column attributes used to create forms and reports.
5. Under Tasks, click **Export**.
 6. For **File format**, select an option:
 - **UNIX** - The resulting file contains rows delimited by line feeds.
 - **DOS** - The resulting file contains rows delimited by carriage returns and line feeds.
 - **Database** - Exports to the file repository instead of producing a file to download.
 7. Click **Export**.

Exporting Team Development Feedback

Use the Export Feedback utility to synchronize feedback from a production or test system to the development system. The development system is identified by using the workspace setting Feedback Synchronization Source Identifier which is set on the Edit Workspace Information page. If exporting from a development system, the export process has an additional attribute called Deployment System where you specify to which deployment system, production or test, you want to synchronize the updated feedback to.

To export Team Development Feedback:

1. Navigate to the Export page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Workspace Utilities**.
 - c. Click **Export**.
2. Click the **Feedback** tab.
3. On the Feedback page:
 - a. **Changes Since** - Select the date of the oldest feedback to export. All feedback from this selected date until the current date is exported.
 - b. **File Format** - Select how rows in the export file are formatted. Options include:
 - **UNIX** - The resulting file contains rows delimited by line feeds.
 - **DOS** - The resulting file contains rows delimited by carriage returns and line feeds.
 - c. Click **Export**.

See Also:

["Importing Team Development Feedback"](#)

Exporting Script Files from the Script Repository

You can transfer selected scripts from your current Script Repository to a Script Repository in a different Workspace by using the Export and Import tasks.

To export script files:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **SQL Scripts**.
3. On the Tasks list, click **Export**.
4. Select the appropriate script files and click **Add to Export**.
5. Review the file name and click **Export All**.

Select the Remove check box to remove the script.



See Also:

"Using SQL Scripts" in *Oracle Application Express SQL Workshop Guide*

Importing Export Files

Once you export an application and any related files, you must import them into the target Oracle Application Express instance and then install them.

Note that you cannot import an application created in a specific release into an earlier version. As a general rule, always import the application first and then the related files.



Tip:

You can simplify the steps needed to deploy an application by creating a custom application.

- [Importing an Application, Page or Component Export](#)
- [About Importing Application Groups](#)
- [Importing a Websheet](#)
- [Importing Plug-ins](#)
- [Importing Themes](#)
- [Importing User Interface Defaults](#)
- [Importing Team Development Feedback](#)

 **See Also:**

- ["How to Create a Custom Application"](#)
- ["How to Move an Application to Another Instance"](#)

Importing an Application, Page or Component Export

To import an Application, Page or Component Export into a target Oracle Application Express instance:

1. Navigate to the Import page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Import**.
2. For Import:
 - a. **Import file** - Navigate to the file.
 - b. **File Type** - Select **Application, Page, or Component Export**.
 - c. Verify that File Character Set is correct.
 - d. Click **Next**.

Once you import a file, you have the option to install it.

3. To install the file, click **Next**.

The Install Application wizard appears.

4. For the Install wizard:
 - a. **Parsing Schema** - Select a schema.

This is the schema against which all of the application's SQL and PL/SQL are parsed.

- b. **Build Status** - Select one of the following:
 - **Run Application Only** - Users can only run an application.
 - **Run and Build Application** - Users can run an application and developers can both run and edit an application.

Selecting **Run Application Only** is an effective way to protect an application from modifications from other developers.

 **Tip:**

If you select **Run Application Only**, the only way to change this setting after you import the application is to log in to Oracle Application Express Administration Services.

- c. **Install As Application** - Select one of the following:
 - **Auto Assign New Application ID**
 - **Reuse Application ID From Export File**

- **Change Application ID**

Use these options to avoid application ID conflicts. These options come in handy when you have two versions of the same application in the same instance. For example, you might be migrating an application to a production instance and still need to maintain the development version.

- d. Click **Install Application**.

**See Also:**

- "Changing Application Build Status Set During Deployment" in *Oracle Application Express Administration Guide*
- "[How to Create a Custom Application](#)"

About Importing Application Groups

Application groups are exported with an application, application exports include an group ID, name, and comments. When importing an application, the application import follows these rules:

- On import, an application uses an existing group if the ID matches.
- If no application group exists with the same ID on the target system, then application import looks for another group ID with the same name. If a name match is found, the application is imported with the ID corresponding to this application group name.
- If no ID or name matches existing application groups on the target system then a new application group is created. The new application group uses the application group ID if that ID is not used in that Oracle Application Express instance. If the group ID is being used, then a new ID is generated.

Importing a Worksheet

To import a Worksheet into a target Oracle Application Express instance:

1. Navigate to the Import page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Import**.
2. For Specify File:
 - a. **Import file** - Navigate to the file.
 - b. **File Type** - Select **Worksheet Application Export**.
 - c. Click **Next**.

Once you import a file, you have the option to install it.

3. To install an imported file, click **Next**.
The Install Application wizard appears.
4. For the Install Application wizard:

- a. **Install As Application** - Select one of the following:
 - **Auto Assign New Application ID**
 - **Reuse Application ID From Export File**
 - **Change Application ID**

Use these options to avoid application ID conflicts. These options come in handy when you have two versions of the same application in the same instance. For example, you might be migrating an application to a production instance and still need to maintain the development version.

- b. Click **Install**.



See Also:

["Creating Worksheet Applications"](#)

Importing Plug-ins

To import an exported plug-in file:

1. Navigate to the Import page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Import**.
2. For Specify File:
 - a. **Import file** - Navigate to the file.
 - b. **File Type** - Select **Plug-in**.
 - c. Verify that File Character Set is correct.
 - d. Click **Next**.

Once you import a file, you have the option to install it.

3. To install an imported file, click **Next**.
4. Click **Install Plug-in**.



Note:

If the plug-in with the same name exists, a dialog displays requesting permission to write over the existing plug-in.

Importing Themes

After you import an application into the target Oracle Application Express instance, you must import all related files.

To import a Theme Export file:

1. Navigate to the Import page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Import**.
2. On Import Definition, select the following:
 - a. **Import file** - Navigate to the file.
 - b. **File Type** - Select **Theme Export**.
 - c. **File Character Set** - Verify that File Character Set is correct.
 - d. Click **Next**.

Once you import a file, you have the option to install it.
3. To install an imported file, click **Next**.
4. Click **Install Theme**.

 **See Also:**

- ["Using Themes"](#)
- ["Exporting Themes"](#)

Importing User Interface Defaults

User Interface Defaults enables you to assign default user interface properties to a table, column, or view within a specified schema.

After you import an application into the target Oracle Application Express instance, you must import all related files.

To import User Interface Defaults:

1. Navigate to the Import page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Import**.
2. Select an application.
3. On Import Definition:
 - a. **Import file** - Navigate to the file.
 - b. **File Type** - Select **User Interface Defaults**.
 - c. **File Character Set** - Verify that File Character Set is correct.
 - d. Click **Next**.

Once you import a file, you have the option to install it.
4. To install an imported file, click **Next**.
5. Click **Install User Interface Defaults**.

 **See Also:**

- "Managing User Interface Defaults" in *Oracle Application Express SQL Workshop Guide*
- ["Exporting User Interface Defaults"](#)

Importing Team Development Feedback

To import Team Development Feedback:

1. Navigate to the Import page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. On the App Builder home page, click **Import**.
2. Select **Feedback**.
3. For **File Character Set**, verify that File Character Set is correct and click **Next**.
Once you import a file, you have the option to install it.
4. To install an imported file, click **Next**.
5. Click **Install Feedback**.

Installing Export Files

After you export an application and any related files and you import them into the target Oracle Application Express instance, the files are stored in the Export Repository. Next, you must install them.

- [About Installing Export Files](#)
- [Accessing the Export Repository](#)
- [Installing an Application Export from the Export Repository](#)
- [Installing Other Files from the Export Repository](#)
- [Deleting Files from the Export Repository](#)
- [Installing Export Files from SQL*Plus](#)
- [Using SQL*Plus to Install Export Files](#)

About Installing Export Files

You can install export files in the following ways:

- After you import the export file, click the **Install** button and follow the on-screen instructions.
- Import the export files into App Builder and then install the files from the Export Repository.
- Install the export files from SQL*Plus.

**Tip:**

You can simplify the steps needed to deploy an application by creating a custom application.

**See Also:**

- ["How to Create a Custom Application"](#)
- ["Managing Exported Files"](#)
- ["Installing Exported Applications into a Runtime Environment"](#) in *Oracle Application Express Administration Guide*

Accessing the Export Repository

When you import an application and any related files into a target Oracle Application Express instance, the files are stored in the Export Repository. You can use the Export Repository to manage archived application exports, imports, and other files.

To access the Export Repository:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Utilities**.
4. Click **Export Repository**.

The Export Repository appears.

Installing an Application Export from the Export Repository

After you import an application export into an Oracle Application Express instance, you must install it before it can become active or available in App Builder.

To install an application export from the Export Repository:

1. Navigate to the Export Repository.
2. Select an application export and click **Install** in the Action column.
3. Specify the following:

- a. **Parsing Schema** - Select a schema.

This is the schema against which all of the application's SQL and PL/SQL are parsed.

- b. **Build Status** - Select one of the following:

- **Run Application Only**
- **Run and Build Application**

Select **Run Application Only** to run the application in the target instance and make it inaccessible to developers.

 **Tip:**

If you select **Run Application Only**, the only way to change this setting after you import the application is to log in to Oracle Application Express Administration Services.

c. **Install As Application** - Select one of the following:

- **Auto Assign New Application ID**
- **Reuse Application ID from Export File**
- **Change Application ID**

Use these options to avoid application ID conflicts. Use these options when you must have two versions of the same application in the same instance. For example, you might be migrating an application to a production instance and still need to maintain the development version.

d. Click **Install**.

 **Tip:**

If you are installing a custom application, the installer prompts you to install the custom application installation scripts. Follow the on-screen instructions.

 **See Also:**

- ["Accessing the Export Repository."](#)
- ["Changing Application Build Status Set During Deployment" in *Oracle Application Express Administration Guide*](#)
- ["How to Create a Custom Application"](#)

Installing Other Files from the Export Repository

After you import files into an Oracle Application Express instance, you must install them before they can become active or available in App Builder.

To install files stored in the Export Repository:

1. Navigate to the Export Repository.
2. Select the file to be installed and click **Install** in the Action column.
3. Follow the on-screen instructions.



See Also:

["Accessing the Export Repository."](#)

Deleting Files from the Export Repository

You can delete a file from the Export Repository.

To delete a file from the Export Repository:

1. Navigate to the Export Repository.
2. Select the file to be deleted and click **Delete Checked**.



See Also:

["Accessing the Export Repository."](#)

Installing Export Files from SQL*Plus

You can also install export files from SQL*Plus. The export file must originate from the same user database account as the one into which you are installing.

The following restrictions apply for export files:

- If the export file is a database application, then the application ID is overwritten. Therefore, the target workspace must own the ID of the application being installed.
- If the export file is a Websheet application, then the target workspace must have Websheet Database objects set-up.
- If the export file is a Websheet application, then the export file must be installed using the Websheet schema in the target workspace.
- If the export file is a Websheet application, then the application ID is overwritten.

Using SQL*Plus to Install Export Files

To install Oracle Application Express export files from SQL*Plus:

1. Log in to SQL*Plus.
2. Run the export file.

For example, if your export file is named f144.sql by default, you would type @f144 at the command prompt.

Managing Application Legacy Components

Although Oracle still supports legacy application components, Oracle does not recommend using them.

- [Managing Legacy Charts](#)
Legacy charts in Oracle Application Express are based on the AnyChart HTML5 Chart Component and the AnyChart Flash Chart Component.
- [Creating a Legacy Master Detail Form](#)
Create a legacy master detail form by running the Create Page Wizard and selecting Legacy Page and then Legacy Master Detail.
- [Managing Legacy Tabular Forms](#)
Create a legacy tabular form by running the Create Page Wizard and selecting Legacy Page and then Tabular Form.
- [Managing Legacy Calendars](#)
Legacy calendars are template-based. In previous releases Legacy Calendar was referred to as Calendar (Template, Classic calendar, Easy calendar, and SQL calendar).
- [Viewing Legacy Components in Gallery](#)
View legacy and unsupported components in the Page Designer gallery by selecting the **Gallery Menu** and then **Show Legacy Components**.

Managing Legacy Charts

Legacy charts in Oracle Application Express are based on the AnyChart HTML5 Chart Component and the AnyChart Flash Chart Component.

Tip:

AnyChart charts are legacy application components. Although Oracle still supports legacy components, Oracle recommends creating new charts based on Oracle JET charts. See "[Creating Charts](#)."

- [About Chart Legacy Types](#)
HTML5 charts and Flash charts support in Oracle Application Express is based on the AnyChart HTML5 Chart Component and AnyChart Flash Chart Component. AnyChart is a flexible Flash and JavaScript-based solution that enables developers to create animated, compact, interactive charts.
- [Creating SQL Queries for Legacy Charts](#)
Use the following syntax when creating legacy charts.
- [Adding a Legacy Chart to a New Page](#)
Add a legacy chart to a new page using the Create Page Wizard.

- [Adding a Legacy Chart in Page Designer](#)
Add a legacy chart to an existing page in Page Designer.
- [Editing Legacy Charts](#)
Edit legacy charts in Page Designer.
- [About Saving Flash Charts](#)
You can add support to save Flash charts locally.
- [Using Custom XML with Flash and HTML5 Charts](#)
Further customize the look and feel of a legacy chart by adding custom XML.

About Chart Legacy Types

HTML5 charts and Flash charts support in Oracle Application Express is based on the AnyChart HTML5 Chart Component and AnyChart Flash Chart Component. AnyChart is a flexible Flash and JavaScript-based solution that enables developers to create animated, compact, interactive charts.

- [About Legacy HTML5 Charts](#)
- [About Legacy Flash Charts](#)

About Legacy HTML5 Charts

HTML5 chart support in Oracle Application Express is based on the AnyChart HTML5 Chart Component. HTML5 charts use a JavaScript chart engine, rendering the chart in SVG format. Flash cannot be rendered on most of the modern mobile devices, however you can now take advantage of our new HTML5 charting solution to incorporate charts in your mobile applications. HTML5 charts are compatible with popular browsers for the following mobile platforms:

- Android: Versions 3.1, 3.2, 4.0, 4.0.3, and 4.1
- IOS (iPhone, iPad, iPod Touch): Safari 3.2 and higher is required



Tip:

To learn more, see <http://6.anychart.com/products/anychart/docs>.

About Legacy Flash Charts

Flash chart support in Oracle Application Express is based on the AnyChart Flash Chart Component. Flash charts are rendered by a browser and require Flash player 9 or later. With AnyChart 6.0, Flash charts are rendered using the FLASH_PREFERRED rendering type. If Flash Player is not available on the device displaying the chart (for example, pages displayed on an iPhone), AnyChart automatically switches to the HTML5 chart engine and displays an SVG-based chart.

Creating SQL Queries for Legacy Charts

Use the following syntax when creating legacy charts.

- [Legacy Chart Syntax](#)

- [Dial Chart Syntax \(Flash and HTML5\)](#)
- [Legacy Multiple Series Syntax \(Flash and HTML5\)](#)
- [Legacy Range Chart Syntax \(Flash and HTML5\)](#)
- [Legacy Scatter Chart Syntax \(Flash and HTML5\)](#)
- [Legacy Candlestick Chart Syntax \(Flash and HTML5\)](#)
- [Legacy Gantt Chart Syntax \(Flash Only\)](#)

Legacy Chart Syntax

When define a chart, you select a chart type and provide a SQL query using the following syntax:

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

Where:

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

For example:

```
SELECT null, last_name, salary
FROM employees
WHERE DEPARTMENT_ID = :P101_DEPARTMENT_ID
```

Dial Chart Syntax (Flash and HTML5)

To create a dial chart, select a dial chart type and provide a SQL query using the following syntax:

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

Where:

- *value* is the starting point on the dial.
- *maximum_value* is the possible highest point on the dial.
- *low_value* and *high_value* are the historical low and high values.

For example:

```
SELECT dbms_random.value(500, 1200), 1300, dbms_random.value(100, 200)
FROM DUAL
```

Legacy Multiple Series Syntax (Flash and HTML5)

For column charts and line charts, you can define multiple series in one SQL query. The series names for these chart types are derived from the corresponding column aliases in the query. To define a multiple series chart, use the following syntax:

```
SELECT link, label, series_1_value [, series_2_value [, ...]]
FROM ...
```

Legacy Range Chart Syntax (Flash and HTML5)

Range charts require two values for each bar. To create a range chart, create a chart and provide a SQL query using the following syntax:

```
SELECT link, label, low_value, high_value
FROM ...
```

Legacy Scatter Chart Syntax (Flash and HTML5)

Legacy scatter charts require an *x* value and *y* value for each point. To create a range chart, create a chart and provide a SQL query using the following syntax:

```
SELECT link, label, x_value, y_value
FROM ...
```

Legacy Candlestick Chart Syntax (Flash and HTML5)

Legacy candlestick charts require open, low, high, and close values for each candlestick. To create a candlestick chart, create a chart and provide a SQL query using the following syntax:

```
SELECT link, label, open, low, high, close
FROM ...
```

Legacy Gantt Chart Syntax (Flash Only)

Project Gantt charts require a task name, task id, parent task id, actual start date, actual_end_date, and progress value for each task. Two optional values for planned start date and planned end date can also be used. To create a Project Gantt chart, create a Flash chart and provide a SQL query using the following syntax:

```
SELECT link, task_name, task_id, parent_task_id, actual_start_date, actual_end_date,
progress
FROM ...
```

Resource Gantt charts require a resource id, resource name, parent resource id, actual start date, and actual end date value for each task. To create a Resource Gantt chart, create a Flash chart and provide a SQL query using the following syntax:

```
SELECT link, resource_id, resource_name, resource_parent_id, actual_start_date,
actual_end_date
FROM ...
```

To represent parent-child hierarchical data on a Resource Gantt chart, provide a SQL query using `START WITH...CONNECT BY` syntax:

```
SELECT link,
       resource_id,
       resource_name,
       resource_parent_id,
       actual_start_date,
       actual_end_date,
FROM ...
START WITH resource_parent_id IS NULL
CONNECT BY PRIOR resource_id = resource_parent_id
ORDER SIBLINGS BY resource_name
```

Adding a Legacy Chart to a New Page

Add a legacy chart to a new page using the Create Page Wizard.

To add a chart on a new page:

 **Note:**

Legacy AnyChart Charts are legacy application components. Although Oracle still supports legacy components, Oracle recommends creating new charts based on Oracle JET Data Visualizations instead.

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).

This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Legacy Page** and then **AnyChart Chart**.

 **Tip:**

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

5. For Chart Type:
 - a. Chart Rendering - Select a chart type.

To learn more about an attribute, see field-level Help.
 - b. Chart Type - Specify a chart type. Depending upon the chart type, you may be prompted to make additional selections.
 - c. Click **Next**.
6. For Chart Type - Select a chart type and click **Next**.
7. For Page and Region Attributes:
 - a. Page Number - Enter a page number on which the chart is to be created.
 - b. Page Name - Enter a page name.
 - c. Page Mode - Select a page mode.

To learn more, see field-level Help.
 - d. Breadcrumb - Select whether to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.

- e. Click **Next**.
8. For Navigation Preference, specify the type of navigation to include on this page and click **Next**. The navigation options (for example, navigation menu or tabs) depends upon the current application theme.
9. For Chart Attributes, specify the appropriate attributes.

 **Tip:**

The attributes that appear depend upon the chart type. To learn more about an attribute, see field-level Help.

10. For Source:

Enter the query that will return the data to display the chart. Depending on the chart type, the required query format is different.

- a. Enter SQL Query or PL/SQL function returning a SQL Query - Specify the SQL query that will populate your chart. For examples, see see field-level Help.
- b. Specify a query by either:
 - Entering a SQL query in the field provided.
 - Clicking the **Build Query** button. When the Build Chart Query Wizard appears, follow the on-screen instructions.
- c. Perform Query validation/Save query without validation - Specify whether or not to validate the chart query. Queries should be validated whenever possible. However when referring to items in your session, this may not possible until running the page, in this case chose to save without validation.
- d. Page Items to Submit - Enter a comma separated list of page items on the current page to be set into session state when the chart data gets read with a separate request.
- e. Maximum Rows - Enter the maximum number of rows you want to use to display the chart.
- f. When No Data Found Message - Enter the text you want to use if the query does not return any rows.
- g. Specify relevant chart attributes.
To learn more about an attribute, see field-level Help.

11. Click **Create**.

 **See Also:**

- ["About Chart Legacy Types"](#)
- ["Creating SQL Queries for Legacy Charts"](#)

Adding a Legacy Chart in Page Designer

Add a legacy chart to an existing page in Page Designer.

To add a legacy chart in Page Designer:

1. View the page to contain the chart in Page Designer.
2. In the Gallery click the Gallery menu and select **Show Legacy Components**.
3. In the Gallery, right-click **Chart - AnyChart [Legacy]**, select **Add To**, and select the appropriate location.

 **Tip:**

In the Gallery, you can also select **Chart - AnyChart [Legacy]** with the mouse and drag it to the appropriate location in the Layout tab.

Page Designer indicates what actions are required next.

The Messages tab displays a red or yellow badge indicating messages you need to address. The Message tab displays two types of messages:

- **Errors** - Error messages display in red. Selecting an error message displays the associated attribute in red in the Property Editor. You must address errors before a page can be saved.
 - **Warnings** - Warning messages display in yellow. Selecting a warning message displays the associated attribute in yellow in the Property Editor. You must address errors before a page can be saved. Note you can save a page without addressing warning messages.
4. Edit the chart.

Charts have three types of attributes: Region attributes, chart Attributes, and Series attributes.

Attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Edit the region attributes.
 - a. In the Rendering tab, locate and select the chart.
The Property Editor displays the chart region attributes
 - b. Under **Identification**, edit the **Title**.
 - c. Review and edit the other attributes.
6. Edit the chart Attributes.

- a. In the Rendering tab, locate the chart and select the **Attributes** node.
 - b. Review and edit the attributes.
7. Edit the Series:
 - a. In the Rendering tab, select the **Series** node.
 - b. Edit the attributes under **Source**.
 - c. Review and edit the other attributes.
8. Click **Save**.



See Also:

"Viewing a Page in Page Designer."

Editing Legacy Charts

Edit legacy charts in Page Designer.

- [Editing Chart Attributes](#)
- [Switching an HTML5 Chart to a Flash Chart](#)
- [Enabling Asynchronous Updates](#)

Editing Chart Attributes

To edit chart attributes:

1. View the page containing the calendar in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

Charts have three types of attributes: Region attributes, chart Attributes, and Series attributes. In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

2. To edit the region attributes.
 - a. In the Rendering tab, locate and select the chart.
The Property Editor displays the chart region attributes

Enabling Asynchronous Updates

Flash and HTML5 charts can monitor information by enabling the Asynchronous Update attribute on the Chart attributes page, or using a dynamic action with the Refresh action.

- [Enabling the Asynchronous Update Attribute](#)
- [Creating a Refresh Dynamic Action in Page Designer](#)

Enabling the Asynchronous Update Attribute

Enabling the Asynchronous Update attribute updates the chart to reflect changes in the underlying data within a specified time interval.

To enable asynchronous updates:

1. View the page in Page Designer.
2. In the Rendering tab, locate the chart and select the **Attributes** node.

Attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

Tip:

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

3. Find the **Automatic Refresh** group:
 - a. For **Automatic Refresh**, select **Yes**.
 - b. For **Interval**, specify the refresh interval, in seconds.
4. Click **Save**.

Note:

"Viewing a Page in Page Designer"

Creating a Refresh Dynamic Action in Page Designer

To create a Refresh dynamic action:

1. View the page in Page Designer.
2. Click the **Dynamic Actions** tab in the left pane.
3. Under Dynamic Actions, right-click **Events** and select **Create Dynamic Action**.

The Messages tab displays a red or yellow badge to identify messages you need to address. Selecting a message displays the associated attribute in the Property Editor. You must address red error message before you can save.

Attributes in Page Designer are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. In the Property Editor, edit the following Dynamic Actions attributes:

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- a. Identification, Name - Enter a name of the dynamic action.
- b. Execution Options, Sequence - Specify the sequence for this component. The sequence determines the order of execution.
- c. When, Event - Specify the event that causes the dynamic action to fire.
- d. When, Selection Type - Select the type of page element or construct to be used to trigger the event. For examples of using a specific selection type, see Help on the associated attribute.

Next, define the action that to be performed if the event evaluates to True.

5. In the Rendering tab, select the first action under **True**. In the Property Editor, edit the following Action attributes:
 - a. Identification, Action - Select **Refresh**.
 - b. Affected Elements, Selection Type - Select **Region**.
 - c. Affected Elements, Region - Select the region containing the chart.
6. Click **Save**.

 **Note:**

"Viewing a Page in Page Designer"

About Saving Flash Charts

You can add support to save Flash charts locally.

To support the saving of an image, AnyChart requires the use of a server-side script that resides on <http://www.anychart.com>. When you attempt to save an image, the data is sent to the server in Base64 encoding. It is then decoded on the server and the decoded response is returned.

If you want to support the saving of charts locally, you must host the decoding script on your local server. The Flash chart would also require you use of the AnyChart JavaScript function `getPNG`. By default, the chart region uses embedded HTML tags

and does not use any AnyChart JavaScript calls. To make use of the AnyChart JavaScript call, you must modify the region source of your Flash charts.

 **See Also:**

"Saving chart as image or vector file" in *AnyChart User's Guide* for more information on adding support to save Flash charts locally.

Using Custom XML with Flash and HTML5 Charts

Further customize the look and feel of a legacy chart by adding custom XML.

To use customize XML to control the look and feel of a legacy chart:

1. View the page in Page Designer.
2. In the Rendering tab, locate the chart and select the **Attributes** node.

Attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

3. Find **Custom XML**:
 - a. Custom - Select **Yes**.
 - b. XML - Enter the custom Chart XML for this chart. Please refer to the *AnyChart XML Reference* for further details:
4. Click **Save**.

 **Tip:**

For more information on supported chart XML formats, see the *AnyChart XML Reference* at:

<http://6.anychart.com/products/anychart/docs/>

Creating a Legacy Master Detail Form

Create a legacy master detail form by running the Create Page Wizard and selecting Legacy Page and then Legacy Master Detail.

Note:

Legacy master detail forms are legacy application components. Although Oracle still supports legacy components, Oracle recommends developers use the new master detail format which includes one or multiple interactive grids. See "[Managing Master Detail Forms](#)."

Developers can create a legacy master detail form using the Create Page Wizard from two tables that do not have foreign key relationship defined in the database. Although Oracle recommends the use of foreign keys and properly defined relationships in the underlying database, developers do not always have the ability to modify their database model.

In a legacy master detail form, the master detail form page includes buttons that enable the user to move forward and backward within the master result set. While running the wizard, you choose which columns determine the navigation order. You also have the option to choose whether to create a two-page or three-page master detail form, with the detail records shown in a report with single-record editing on a separate page.

To create a legacy master detail form using the create page wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Legacy Page** and then **Legacy Master Detail**.

Tip:

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

5. On Master Table and Columns:
 - a. Table/ View Owner- Choose the schema that owns the master table.
 - b. Table/ View Name - Select the table or view which contains the columns to be included in the master page.

- c. Select Columns- Select the columns to display in your master page. You may add additional columns later.
 - d. Use User Interface Defaults -Select whether to use User Interface Default values to create the form. If you select **No**, the wizard does not use User Interface Default values.
 - e. Click **Next**.
6. On Detail Table and Columns:
 - a. Show Only Related Tables - Specify whether to restrict the list of detail tables to only those tables related to the master table. Select **Yes** or **No**.
 - b. Table / View Owner - Select the table or view owner. The default selected owner for the table or view is the same owner as for your application. If the owner of your application has select privileges on tables or views in other schemas, you may select another schema.
 - c. Table/View Name - Select the table or view which contains the columns to be included in the detail page.
 - d. Select Columns - Select the columns to display in your detail page. You may add columns later.
 - e. Click **Next**.

Next, define a primary key. Forms perform insert, update and delete operations on table rows in the database. You identify the rows by using either a primary key defined on the table, or the ROWID pseudo column, which uniquely identifies a row in a table.
7. On Define Primary Key:

Select one for each table.

 - a. **Managed by Database (ROWID)** - Use the ROWID pseudo column to identify rows to update and delete.
 - b. **Select Primary Key Column(s)** - Use the source table's primary key column. If selected, specify Primary Key Column 1 and optionally Primary Key Column 2. See field-level Help for more information.
 - c. Click **Next**.
8. On Master Detail Link:

Define the relationship between the Master and Detail tables. You may type in the column names (fully qualified with table names), or choose them from the list.

 - a. Link Master Column - Select the master column.
 - b. Link Detail Column - Select the detail column.
 - c. Click **Next**.
9. On Master Options:

If a navigation order column is not defined, the master update form will navigate by the primary key column. By default, this wizard creates a master report page. You can choose to not create master report page if you already have a report page.

 - a. Include master row navigation? - Select **Yes** or **No**. If you select **Yes**, define the navigation order columns. If a navigation order column is not defined, the master update form will navigate by the primary key column.

If a navigation order column is not defined, the master update form will navigate by the primary key column. By default, this wizard creates a master report page. You can choose to not create master report page if you already have a report page.

- b. Master Row Navigation Order- Select the master row navigation order.
 - c. Secondary Navigation Order - Select the secondary navigation order.
 - d. Include master report? - Click **No** to if you already have a report page.
 - e. Click **Next**.
10. On Layout:
- a. Edit detail as tabular form on same page - Select to create a two page master detail.
 - b. Edit detail on separate page - Select to create three page master detail.
 - c. Click **Next**.
11. On Page Attributes:

**Note:**

This page specifies master and detail page information. If the pages you specify do not exist, the pages will be created for you.

Master Page

- a. Page Number - Enter the page numbers that will display master and detail. A new page will be created if one does not already exist.
- b. Page Title - Enter page titles for the new pages. If the pages already exist, the existing page names display.
- c. Region Title - Enter names for the regions containing the master and detail.
- d. Page Mode - Identify the page mode, See field-level Help for more information.

Detail Page

- a. Page Number - Enter the page number.
 - b. Page Title - Enter the page title.
 - c. Click **Next**.
12. On Navigation Menu, select how you want this page integrated into the Navigation Menu and click **Next**. See field-level Help for more information.
13. On Confirm:
- Verify the attributes and confirm your selections.
14. Click **Create**.



See Also:

["Managing Master Detail Forms"](#)

Managing Legacy Tabular Forms

Create a legacy tabular form by running the Create Page Wizard and selecting Legacy Page and then Tabular Form.



Note:

A tabular form is a legacy application component. Although still supported, Oracle recommends developers create interactive grids instead.

A legacy tabular form enables users to update multiple rows in a table at once from a single page. When created using a wizard, a legacy tabular form enables you to perform update, insert, and delete operations on multiple rows in a database table. Tabular forms include a built-in multiple row update process that performs optimistic locking behind the scenes to maintain the data integrity.

- [About Creating Tabular Forms](#)
- [About Running a Tabular Form](#)
- [How Tabular Forms Work](#)
- [Creating a Legacy Tabular Form](#)



See Also:

- ["Upgrading an Application to Include New Components"](#)
- ["Managing Interactive Grids"](#)
- ["Developing Forms"](#)

About Creating Tabular Forms

Tabular forms created using wizards include automatic DML processes. These wizards create a form that performs update, insert, and delete operations on multiple rows in a database table. Alternatively, you can build a tabular form manually either by creating a updatable report region or standard report region with updatable columns with custom PL/SQL page processes to process the tabular form data.

Best Practices When Creating Tabular Forms

Oracle recommends the following best practices when creating tabular forms:

- Do not modify of the select list of a SQL statement of a tabular form after it has been generated using a wizard. If you do modify the query, make sure the values of the updatable columns are not altered after being queried from the database by the Application Express engine.
- Do not add custom JavaScript to display types that use display type specific JavaScript logic (for example, radio groups, simple check boxes, and popup key LOVs).

About Running a Tabular Form

When running a tabular form, data is rendered as a report having updatable columns shown using various form elements (including text fields, text areas, date pickers, select lists, radio groups, and so on).

Tabular Form

| <input type="checkbox"/> | Project | Task Name | Start Date | End Date | Status |
|--------------------------|----------------------|-------------------------|------------|-----------|---------|
| <input type="checkbox"/> | APEX Environment Cc | Identify server require | 14-JAN-15 | 15-JAN-15 | Closed |
| <input type="checkbox"/> | Maintain Support Sys | HR software upgrade | 14-JAN-15 | 11-MAR-15 | On-Hold |
| <input type="checkbox"/> | Maintain Support Sys | Apply Billing System | 14-JAN-15 | 15-MAR-15 | On-Hold |
| <input type="checkbox"/> | APEX Environment Cc | Determine Web lister | 15-JAN-15 | 15-JAN-15 | Closed |
| <input type="checkbox"/> | APEX Environment Cc | Specify security auth | 16-JAN-15 | 18-JAN-15 | Closed |
| <input type="checkbox"/> | APEX Environment Cc | Select servers for Dev | 16-JAN-15 | 21-JAN-15 | Closed |
| <input type="checkbox"/> | Email Integration | Complete plan | 21-JAN-15 | 26-FEB-15 | Closed |
| <input type="checkbox"/> | APEX Environment Cc | Configure Workspace | 23-JAN-15 | 23-JAN-15 | Closed |
| <input type="checkbox"/> | APEX Environment Cc | Create pilot workspac | 23-JAN-15 | 23-JAN-15 | Closed |
| <input type="checkbox"/> | Maintain Support Sys | Arrange for vacation | 04-MAR-15 | 27-MAR-15 | Open |

row(s) 1 - 10 of 73 [Next](#)

[Add Row](#)

[Cancel](#) [Delete](#) [Apply Changes](#)

By default, date columns are created using the Date Picker item type. If user interface defaults are defined, other columns can be automatically created as select lists based on lists of values, or numeric fields and large text areas.

For numeric and date fields, you can also pre-define date and number format masks, or apply those format masks after generating the initial form.

Each report also includes a check box column (called a row selector). Users use this row selector to select one or more columns and update or delete data. To add a row, users click **Add Row**.

 **Tip:**

You can control the appearance of a specific column by editing column attributes.

 **See Also:**

"Managing User Interface Defaults" in *Oracle Application Express SQL Workshop Guide*

How Tabular Forms Work

Tabular forms are based on a SQL query in which the query columns are tied to the underlying table columns. Unlike a single record form, tabular forms do not store data in session state. When a tabular form page is submitted, the tabular form data is processed using built-in data manipulation language (DML), or a custom PL/SQL page processes.

A tabular form with automatic DML has two processes for multi-row operations:

- A Multi Row Update (MRU) process performs create and update operations.
- Multi Row Delete (MRD) process deletes requests on one or more rows.

MRU and MRD processes reference the underlying table name and the primary key columns. The primary key columns must be part of the tabular form SQL query, and the report columns must be either hidden or display-only (save state) in order for the DML processes to be able to identify the correct records. For new records, the primary key values must be `NULL` to be identified as new records. Delete operations are performed by referencing row number of the row to be deleted. Users can check one or more rows on the current page and when they click the Delete button, the row numbers identify the corresponding primary key value(s) and the matching rows are deleted.

Wizard generated tabular forms built-in DML include a checksum column to prevent lost updates. Using a method called optimistic locking, a checksum is computed for each row when rendering a tabular form. This checksum is stored in an extra hidden tabular form column. Upon submission of the form for update, an additional set of checksums is computed on the then current data stored in the database. The new checksums are compared to those built when the tabular form was first rendered, and if there are any mismatches, the update is rejected, because the data in the database was either modified by another user or another process in the time between the tabular form was rendered and the time the data was submitted.

Wizard generated tabular forms also automatically create validations for some columns. Validations are created for columns that are set to `NOT NULL` in the underlying table and columns of type `NUMBER`, `DATE` or `TIMESTAMP`. Note that validations are not

created for columns if the column is set to read-only, either based on the user's selection, or defined user interface defaults.

Creating a Legacy Tabular Form

Use the Create Page Wizard to add a legacy tabular form on a new page. Once you provide a table name on which to build the tabular form, the wizard reads the table columns from the database data dictionary. When creating a tabular form, Oracle recommends using a generic column template and avoiding column report templates.

Note:

Legacy tabular forms are legacy application components. Although Oracle still supports legacy components, Oracle recommends creating interactive grids instead.

To create a legacy tabular form using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page (optional).
This attribute only displays for applications using older themes and for which Desktop and Mobile User Interfaces have been defined.
 - b. Page Type - Select **Legacy Page** and then **Tabular Form**.

Tip:

Component pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

5. For Page Attributes:
 - a. Page Number - If you identify a new page number, the wizard creates a new page. If you identify an existing page number, the wizard adds the component to that page.
 - b. Page Name - Specify a name for the page.
 - c. Page Mode - Identify the page mode. Options include:
 - **Normal** - The page is presented as a normal Application Express application page.
 - **Modal Dialog** - The page is presented as a modal dialog. A modal dialog is an overlay window positioned within the viewport, which remains active and focused until the end user dismisses (closes) it. The underlying page

is grayed out and the end user is prevented from interacting with the rest of the page until the dialog is closed.

- d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page. If you elect to include breadcrumb navigation, additional attributes appear.
 - e. Click **Next**.
6. For Navigation Preference:
 - a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
 - b. Click **Next**.
7. For Tabular Form Source:
 - a. Table/View Owner - Select the owner of the table on which you are building a report.
 - b. Table/View Name - Select the table or view on which the form will be based.
 - c. Select Columns - Select one or more columns to be included in the tabular form. In the pages that follow you define which columns should be updatable.
 - d. Click **Create**.



See Also:

- ["Report Templates"](#)
- ["Managing Interactive Grids"](#)

Managing Legacy Calendars

Legacy calendars are template-based. In previous releases Legacy Calendar was referred to as Calendar (Template, Classic calendar, Easy calendar, and SQL calendar).



Tip:

Legacy calendars are legacy application components. Although Oracle still supports legacy components, the creation of new calendars is based on the FullCalendar jQuery library.

- [Editing Calendar Attributes in the Property Editor](#)
- [About Supported Calendar Substitution Strings in Legacy Calendars](#)

Editing Calendar Attributes in the Property Editor

To edit calendar Attributes:

1. View the page containing the calendar in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.

Page Designer appears.

In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

 **Tip:**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

2. To edit the region attributes.
 - a. In the Rendering tab, locate and select the calendar.
 - b. Review and edit the region attributes in the Property Editor.
3. To edit the calendar Attributes.
 - a. In the Rendering tab, locate the calendar and select the **Attributes** node.
 - b. Review and edit the calendar Attributes in the Property Editor.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

About Supported Calendar Substitution Strings in Legacy Calendars

App Builder supports many date format substitution strings. You can view a complete list of supported substitution strings on the Calendar Templates page.

To view a list of supported substitution strings for calendars:

1. Navigate to the appropriate calendar template.
2. View the Substitution Stings list on the right side of the page.

 **See Also:**

["Using Themes and Theme Styles"](#)

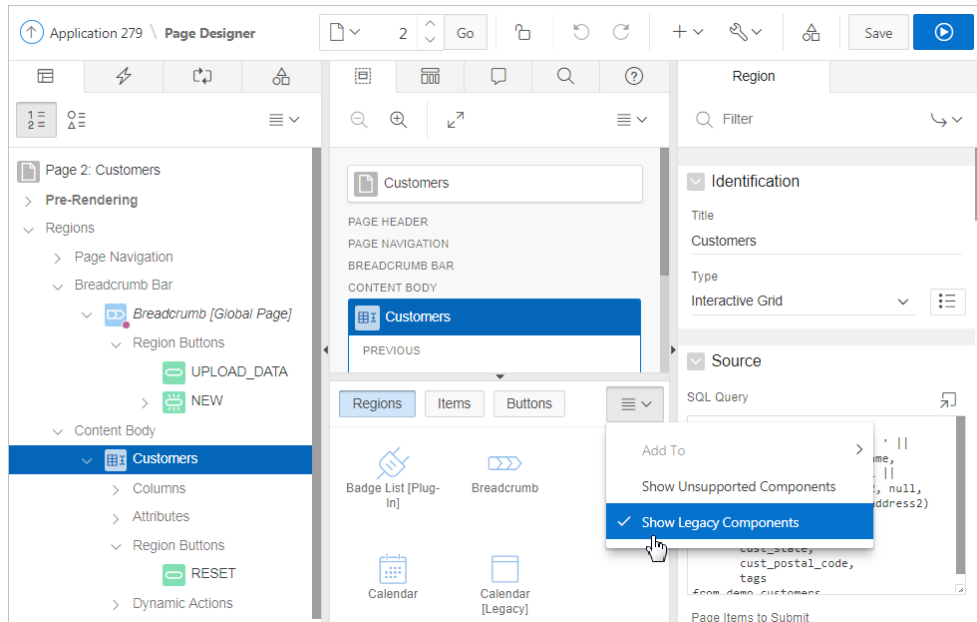
Viewing Legacy Components in Gallery

View legacy and unsupported components in the Page Designer gallery by selecting the **Gallery Menu** and then **Show Legacy Components**.

To view legacy components in the gallery:

1. Navigate to the Workspace home page.

2. Click the **App Builder** icon.
 3. Select an application.
 4. Select a page.
- The page appears in Page Designer.
5. Click **Gallery Menu** and select **Show Legacy Components**.



A

About Item Types

Learn about key attributes that control item functionality.

An item is part of an HTML form. An item can be a text field, text area, password, select list, check box, and so on. Developers can control how a page item works by editing the page item attributes. To learn more about specific item attributes, see field-level Help

- [Available Item Types](#)



See Also:

- ["Viewing Field-Level Help"](#)
- ["Understanding Page-Level Items"](#)
- ["Editing Page-Level Items"](#)

Available Item Types

The following table describes available item types in App Builder and notes important attributes that control item behavior.

Table A-1 Available Item Types

| Item Type | Description |
|--------------|--|
| Check Box | <p>Displays multiple values as check boxes, enabling the end user to select multiple values. A list of values is required for items displayed as check boxes. The values corresponding to the checked boxes are stored in a single colon-delimited string.</p> <p>Note: For Yes/No check boxes (that is, a checkbox with only one value), Oracle recommends using a Switch item type.</p> <p>The following example displays employee names (ename), but returns employee numbers (empno):</p> <pre>SELECT ename, empno FROM emp;</pre> <p>See Also: "APEX_UTIL" in <i>Oracle Application Express API Reference</i> for information about breaking up returned values</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none">• Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details.• Settings, Number Columns - Enter the number of layout columns used to display the check boxes. For example, a value of 2 would display two columns of check boxes. By default, all values display in one column. <p>See Also: "Creating a Cascading List of Values"</p> |
| Color Picker | <p>Displays a text field with a color picker icon. End users can either enter a HTML color code directly, or click the icon to display a color palette popup. The end user can click a color from the palette which returns the corresponding HTML color code for the selected color.</p> <p>Note: When creating a new item of this type, first select Popup List of Values and then Color Picker.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none">• Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. <p>See Also: "Configuring Color Picker Display"</p> |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|---------------|---|
| Date Picker | <p>Displays a text field with a calendar icon. End users can either enter the date directly into the text item, or click on the calendar icon to select a date, and optionally a time, from the calendar popup.</p> <p>Time is only displayed in the calendar popup if the Format Mask for this item includes time components.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Highlighted Date - Enter a default date value to be highlighted when the calendar pop-up displays. See Help for more details. • Settings, Minimum Date - Enter the minimum date that can be input or selected. • Settings, Maximum Date - Enter the maximum date that can be input or selected. • Settings, Show - Select when the date picker pop-up calendar displays. • Settings, Show other Months - Specify whether the leading dates from the previous month and the trailing dates for the next month display. The leading and trailing dates fully populate the first and last lines of the calendar. • Settings, Navigation List for - Select whether select lists are provided within the calendar pop-up for the month and/or year. • Appearance, Format Mask - Enter a date format mask to apply to this component. • Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. |
| Display Image | <p>Displays an image stored in a database BLOB columns, or based on an image URL.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Based On - Determines where the item is retrieved from. To learn more about an attribute, see field-level Help. • Settings, Alternative Text Column - Enter the uppercase page or application item name which holds the alternative text for this image. This text is read by screen readers when an end user navigates to this item. • Settings, Filename Column - Enter the uppercase page or application item name that holds the filename column value associated with the BLOB column. • Settings, BLOB Last Updated Column - If specified, the column is used for browser caching. It identifies if the image changed since the last browser request. If not entered, no caching is done. Note: This attribute is case sensitive and the column must be of type DATE. |
| Display Only | <p>Displays a non-enterable text item.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Save Session State - Stores the current value in session state when the page gets submitted. • Settings, Based On - Specifies what gets displayed in the field. Options include: <ul style="list-style-type: none"> – Page Item Value - The current value of the page item displays. – Display Value of List of Values - The current value of the page item maps to a different display value using a list of values. – Output of PL/SQL Code - The specified PL/SQL code takes care of emitting the necessary output for the page item. <p>Settings, Show Line Breaks - Select Yes to convert line breaks in the display value to an HTML line break and display them in HTML output.</p> |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|-------------|--|
| File Browse | <p data-bbox="444 338 1373 394">Displays a text field with a Browse button. The Browse button enables the user to locate and upload a file from a local file system.</p> <p data-bbox="444 405 878 430">See Also: "About Securing File Uploads"</p> <p data-bbox="444 441 716 466">Page Designer attributes:</p> <ul style="list-style-type: none"> <li data-bbox="444 476 1373 533">• Settings, Storage Type - Identify where the uploaded file is stored. Options include: <ul style="list-style-type: none"> <li data-bbox="492 533 1373 678">– BLOB column specified in Item Source attribute - Stores the uploaded file in the table used by the Automatic Row Processing (DML) process and the column specified in the item source attribute. The column has to be of data type BLOB. If the file gets downloaded, the table name of the Automatic Row Fetch process is used. <li data-bbox="492 688 1373 833">– Table APEX_APPLICATION_TEMP_FILES - Stores the uploaded file(s) in a temporary location that you can access with the view APEX_APPLICATION_TEMP_FILES. Application Express will automatically delete the file(s) at the end of the session or at the end of the upload request, depending on what you choose for Purge File At. <p data-bbox="492 844 1216 869">If Storage Type equals Table APEX_APPLICATION_TEMP_FILES:</p> <ul style="list-style-type: none"> <li data-bbox="492 879 1373 936">– Purge File at - Define when Application Express should remove the temporary file. <li data-bbox="492 947 1373 1003">– Allow Multiple Files - Specify whether multiple files can be uploaded at once. Multiple file names will be stored as a colon delimited list in the page item.. <li data-bbox="492 1014 1373 1071">– File Types - Specify a comma delimited list of file types which might be used by the browser to restrict which files can be uploaded. <p data-bbox="492 1081 1187 1106">If Storage Type equals BLOB column specified in Item Source:</p> <ul style="list-style-type: none"> <li data-bbox="492 1117 1373 1226">– MIME Type Column - Enter the uppercase page or application item name used to store the mime type of the uploaded file. The mime type is used by the browser to identify which application is used to display the content of the BLOB when the file is downloaded. <li data-bbox="492 1236 1373 1293">– Filename Column - Enter the page or application item used to store the filename of the uploaded file. . <li data-bbox="492 1304 1373 1413">– Character Set Column - Enter the uppercase page or application item name used to store the character set of the uploaded file. The character set of the BLOB is not automatically set on upload. End users should set the character set for the document they are uploading. <li data-bbox="492 1423 1373 1497">– BLOB Last Updated Column - Enter the uppercase page or application item name used to store the current timestamp when a file is uploaded. This date value is used for browser caching.. <li data-bbox="492 1507 1373 1564">– Display Download Link - Specify whether the download link displays when there is a non-empty BLOB associated with this item. <li data-bbox="492 1575 1373 1684">– Download Link Text - Enter the text that displays as part of the download link. If not specified, the text stored in the Text Message for APEX.FILE_BROWSE.DOWNLOAD_LINK_TEXTwithin Shared Components, displays. <li data-bbox="492 1694 1373 1751">– Content Disposition - Select how the browser handles the content when downloading. <ul style="list-style-type: none"> <li data-bbox="444 1761 1373 1785">• Settings, File Types - Specify a comma delimited list of file types which might be used by the browser to restrict which files can be uploaded. |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|---------------|---|
| Hidden | <p>Items that are included within the page source but are not rendered. Hidden item values are saved in session state. They are generally used to store values required by page processing or other page items, but should not be displayed to the end user.</p> <p>Page Designer attribute:</p> <ul style="list-style-type: none"> • Settings, Value is Protected - Select Yes to prevent hidden values from being manipulated when a page is posted. |
| List Manager | <p>Displays a text item with a popup list of values icon, Add and Remove buttons, and a list of selected values.</p> <p>You can type in the value or pick from the list of available items. You can then utilize the buttons to manage the values selected. The selected values are stored in a single colon-delimited string.</p> <p>Page Designer attributes:</p> <ul style="list-style-type: none"> • Settings, Fetch - Select how data is displayed when the Popup LOV is opened. • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. <p>See Also: "Creating a Cascading List of Values"</p> |
| Number Field | <p>Displays a number field. You can configure number range checks for values entered by configuring the Minimum Value and Maximum Value attributes. This item type automatically checks if the entered value is a number. No extra validations are required.</p> <p>Page Designer attributes:</p> <ul style="list-style-type: none"> • Settings, Minimum Value - Enter the minimum value permitted. The value can be a static value or a dynamic value using substitution syntax. • Settings, Maximum Value - Enter the maximum value permitted. The value can be a static value or a dynamic value using substitution syntax. • Settings, Number Alignment - Select how to align the number that displays. This setting does not apply when the item is rendered as Read Only or where HTML Form Element Attributes are specified. • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. |
| Password | <p>Displays an HTML password form element. As the end user enters text a black dot is displayed for that character, instead of the actual character entered.</p> <p>Page Designer attributes:</p> <ul style="list-style-type: none"> • Settings, Submit when Enter Pressed - Specify whether pressing the Enter key while in this field automatically submits the page. • Settings, Does not save state - Specify whether the password is saved in session state. <p>Oracle strongly recommends that you do not write the password into session state, for security reasons. The value can potentially be read from browser cache and Application Express session tables. If you do set this attribute to No, consider setting Store value encrypted in session state to Yes.</p> <ul style="list-style-type: none"> • Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. <p>See Also: "About Session State and Security".</p> |
| Percent Graph | <p>Displays the value as a percentage graph. The value retrieved must be between 0 and 100.</p> |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|-------------|--|
| Popup LOV | <p>Displays a text item with a popup list of values icon. When the end user clicks the icon, a popup window appears with a search field, and a list of supported values.</p> <p>Popup LOV is best suited for large lists since end users can enter search criteria to reduce the available values displayed. For relatively small lists Select List is often better suited.</p> <pre data-bbox="444 512 769 562">SELECT ename name, empno id FROM emp</pre> <p>If one of the columns is an expression, remember to use an alias. For example:</p> <pre data-bbox="444 646 1065 667">SELECT ename ' ' job display_value, empno FROM emp</pre> <p>Page Designer item attributes:</p> <ul data-bbox="444 737 1370 884" style="list-style-type: none"> • Settings, Input Field - Select whether the item is enterable. • Settings, Fetch - Select how data is initially fetched for the list of values. • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. <p>See Also: "Creating a Cascading List of Values"</p> |
| Radio Group | <p>Displays multiple values as radio group options, enabling the end user to select a single value.</p> <p>The following example displays employee names (ename), but returns employee numbers (empno):</p> <pre data-bbox="444 1087 769 1108">SELECT ename, empno FROM emp</pre> <p>Page Designer item attributes:</p> <ul data-bbox="444 1178 1370 1650" style="list-style-type: none"> • Settings, Number of Columns - Enter the number of radio group columns to display. For example, a value of 2 would display two columns. If there were five values then it would display over three rows. • Settings, Page Action on Selection - Select what action is taken when a radio group value is selected. See Help for more details. • Settings, Execute Validations - Displays if Page Action when Value Changed is set to Submit Page. Controls the execution of validations when a page is submitted. Set this attribute to No to prevent the execution of validations and associated error messages. Set this attribute to Yes to execute all page and built-in validations and associated error messages when the page is submitted. Use this attribute in conjunction with the validation attribute, Always Execute. To learn more, see "About Determining When Validations Execute." <p>See Also: "Creating a Cascading List of Values"</p> <ul data-bbox="444 1570 1370 1650" style="list-style-type: none"> • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|------------------|--|
| Rich Text Editor | <p>Displays a text area with comprehensive text formatting options. End users can enhance the content displayed in a similar fashion to using a word processor, such as Microsoft Word. End users can also cut and paste documents that are in rich text format (RTF) file format into this item.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Editor - Select the editor used to display the text formatting options available. • Settings, Toolbar - Select the number of text formatting options available within the editor. • Settings, Toolbar Expanded - Specify whether the editor is initially expanded within the text area. • Settings, Skin - Select the editor skin used to style the editor. • Settings, Toolbar Position - Specify where in the text area the editor displays. • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. |
| Select List | <p>Displays an item with a built-in list of values selector. When the end user clicks the item, the list of supported values displays directly inline with the current item.</p> <p>Select List is best suited for relatively small, discrete lists. End users can very quickly select a value from the list without needing to change focus to a popup dialog. For large lists, consider using a Popup LOV.</p> <pre data-bbox="521 1010 786 1150">select [displayValue], [returnValue] from ... where ... order by ...</pre> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Page Action on Selection - Select what action is taken when a list value is selected. See help for options. • Settings, Execute Validations - Displays if Page Action on Selection is set to Submit Page. Controls the execution of validations when a page is submitted. Set this attribute to No to prevent the execution of validations and associated error messages. Set this attribute to Yes to execute all page and built-in validations and associated error messages when the page is submitted. Use this attribute in conjunction with the validation attribute, Always Execute. To learn more, see "About Determining When Validations Execute." • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|-----------|--|
| Shuttle | <p>Displays as a multiple select list that includes two boxes containing lists. The left list displays the source list of values that have not been selected and the right list shows the currently selected values. End users can select one or more values in a list then use the shuttle controls to move the selected values or all values.</p> <p>The current values are stored in a single colon-delimited string.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Show Controls - Select what data movement controls displays in the middle of the shuttle. Options include: None, Moving Only, Ordering Only, or All. • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. <p>Note: To create this item type, you must define a list of values. See "Creating Lists of Values at the Application-Level," "Working with Multiple Select List Item," and "Creating a Cascading List of Values."</p> |
| Switch | <p>Displays as a flip toggle switch.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Settings - Select whether application level component settings are used, or if custom settings are used for this item. Options include: Use Component Settings or Custom. If Settings is set to Custom, the additional attributes display. • Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. <p>See Also: "Configuring Switch Defaults"</p> |
| Slider | <p>Only applicable for jQuery Mobile applications. Displays a slider item type, which enables users to update the item value by sliding the control left or right.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Minimum Value - Enter the minimum value permitted. The value can be a static value or a dynamic value using substitution syntax. • Settings, Maximum Value - Enter the maximum value permitted. The value can be a static value or a dynamic value using substitution syntax. • Settings, Step Increment - Enter a value to force the slider to snap to a specific increment. The value can be a static value or a dynamic value using substitution syntax. • Settings, Highlight Selected - Specify whether the slider highlights the track up to the current slider handle position. • Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|------------------------------|---|
| Text Field | <p>Displays the item as a text field.</p> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> Settings, Subtype - Select the HTML5 text subtype. This enables devices with on-screen keyboards to show an optimized keyboard layout specific to the subtype, for easier data input. The subtype selection is also used to render an appropriate link with the value of the page item, if it is rendered read only. This attribute does not change the text item, or data entry, when using a physical keyboard. <p>Note: This HTML5 feature only works in modern browsers. Older, non HTML5-compliant, browsers ignore this attribute and render the page item as a normal text field.</p> Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. |
| Text Field with autocomplete | <p>Displays a text field that displays a list of possible values, based on the text already entered by the end user, inline with the text item. The list is further refined as the end user types in more text.</p> <pre data-bbox="524 852 699 961">select [column] from ... where ... order by ...</pre> <p>Page Designer item attributes:</p> <ul style="list-style-type: none"> Settings, Search - Select how the search against the entered value is performed. When using Lazy Loading this setting can have an impact on query performance. Selecting Contains & Case Sensitive or Contains & Ignore Case prevents the use of column indexes. Settings, Lazy Loading - Specify whether matching values are retrieved from the database using Lazy Loading techniques. Lazy loading retrieves matching records from the database each time the user types a character. Otherwise, all of the list values are loaded on page load. Oracle recommends using Lazy Loading when you have a large select list, as loading a large number of records may impact the time to initially load the page. Settings, Maximum Values in List - Enter the maximum number of items to display in the select box. This value can help avoid performance problems by returning too many rows, and also prevent a very large return set when only a few letters have been entered. Settings, Automatic Fill - Specify whether the text field is filled automatically with the first autocomplete value. See Help for more details. Settings, Select First - Specify whether the first autocomplete value is automatically selected when the end users hits tab/return, even if it has not been handpicked by keyboard or mouse action. If there is a handpicked (highlighted) result, that result takes precedence. Settings, Highlight Search - Specify whether the matching text is highlighted in the select box of matching entries. Validation, Value Required - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. |

Table A-1 (Cont.) Available Item Types

| Item Type | Description |
|-----------|--|
| Text Area | <p data-bbox="444 338 797 363">Displays a multiple-row text area.</p> <p data-bbox="444 373 1365 457">Depending on your application configuration, you may also have additional item type plug-ins available here. These display as <My Plug-in> [Plug-in]. The following plug-ins install into this application's shared components:</p> <p data-bbox="444 468 764 493">Page Designer item attributes:</p> <ul data-bbox="444 504 1365 800" style="list-style-type: none"><li data-bbox="444 504 1365 562">• Settings, Resizable - Specify whether the text area resizing using the mouse is enabled.<li data-bbox="444 573 1365 646">• Settings, Auto-Height - Specify whether the height of the text area varies based on the amount of text displayed. Use this option to scale the text area to the amount of data.<li data-bbox="444 657 1365 709">• Settings, Character Counter - Select Yes to include a counter that displays the number of characters entered in the field.<li data-bbox="444 720 1365 800">• Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. |

B

Oracle Application Express Limits

Learn about limits associated with Oracle Application Express components.

The Oracle Application Express development environment has no limits as to the number of workspaces or number of users that can exist in a given workspace.

- [Component Limits](#)

Component Limits

The following table describes limits associated with Oracle Application Express components, such as interactive reports, classic reports, forms, and tabular forms.

Table B-1 Component Limits

| Component | Limits | To Learn More... |
|---------------------|--|---|
| Interactive reports | 999 rows per column heading filter (if no custom LOV is specified in the column attributes). | "Developing Reports" |
| Forms | 32767 bytes for a text area or rich text editor item. Two columns for primary key (when using built in DML processes). For users with more than two primary key columns defined, you can also specify a ROWID. | "Developing Forms" |
| Tabular Forms | One wizard-generated tabular form per page (using built-in DML). 50 editable tabular form columns (apex_application.g_f01-apex_application.g_f50), generated with apex_item or the built-in tabular form display types. | "Managing Legacy Tabular Forms" |
| Item names | Item names should not exceed 30 characters. Items longer than 30 characters cannot be referenced using bind variable syntax. | "Referencing Session State Using Bind Variable Syntax" "About Item Naming Conventions" |
| Validations | Text entered for validations may not exceed 3,950 characters. | "Understanding Validations" |
| Data loading | When loading data, the maximum number of columns is 45. | "Creating Applications with Data Loading Capability" |

C

Available Conditions

Learn about conditions available in App Builder.

- [What Are Conditions?](#)
- [Conditions Available in App Builder](#)

What Are Conditions?

A condition is a small unit of logic that helps developers control the display of regions, items, buttons, and tabs as well execute processes, computations and validations. When you apply a condition to a control or component, the condition is evaluated. Whether a condition passes or fails determines whether a control or component displays, or page processing executes.

You can specify conditions by selecting a condition type when you create the control (region, button, or item) or component (tab, list, or navigation bar), or by making a selection under the condition attribute.



See Also:

["Understanding Conditional Rendering and Processing"](#) and ["Conditions Available in App Builder"](#)

Conditions Available in App Builder

The following table describes many App Builder conditions. To view a complete listing of all available conditions for a given control or component, expand the condition type list.

Table C-1 Available Conditions

| Condition | Description |
|----------------------|---|
| Rows returned | The SQL query returns one or more rows. |
| No Rows returned | The SQL Query returns no rows |
| SQL Expression | The SQL expression evaluates to TRUE. |
| PLSQL Expression | The SQL expression evaluates to TRUE. |
| PL/SQL Function Body | The PL/SQL function body returns TRUE. |
| Request = Value | The page request is equal to the text you enter into the Value attribute. |

Table C-1 (Cont.) Available Conditions

| Condition | Description |
|--|---|
| Request != Value | The page request is not equal to the text you enter into the Value attribute. |
| Request is contained in Value | The page request is contained in the text you enter into the Value attribute. |
| Request is NOT contained in Value | The page request is not contained in the text you enter into the Value attribute. |
| Item = Value | The value of the selected item is equal to the text you enter into the Value attribute. |
| Item != Value | The value of the selected item is not equal to the text you enter into the Value attribute. |
| Item is NULL | The value of the selected item is empty. |
| Item is NOT NULL | The value of the selected item is not empty. |
| Item is zero | The value of the selected item is the number zero. |
| Item is NOT zero | The value of the selected item is not the number zero. |
| Item is NULL or zero | The value of the selected item is empty or the number zero. |
| Item is NOT NULL and NOT zero | The value of the selected item is not empty and not the number zero. |
| Item contains no spaces | The value of the selected item has no spaces. |
| Item is numeric | The value of the selected Item is numeric. |
| Item is NOT numeric | The value of the selected item is not numeric. |
| Item is alphanumeric | The value of the selected item is alphanumeric, containing only letters or numbers and no special characters. |
| Item is in colon delimited list | The value of the selected item is completely contained in the text you enter into the Value attribute. |
| Item is NOT in colon delimited list | The value of the selected Item is not completely contained in the text you entered into the Value attribute. |
| User Preference = Value | The value of the Preference entered is equal to the text you enter into the Value attribute. |
| User Preference != Value | The value of the Preference entered is not equal to the text you enter into the Value attribute. |
| Current Page = Page | The current page is equal to the value you enter into Page. |
| Current Page != Page | The current page is not equal to the value you enter into Page. |
| Current Page is in comma delimited list | The current page is in the comma separated list you enter into Pages. |
| Current Page is NOT in comma delimited list | The current page is not in the comma separated list you enter into Pages. |
| Current page is in Printer Friendly mode | The current page has been toggled to 'Printer Friendly Mode' by the end user. |
| Current page is NOT in Printer Friendly mode | The current page has not been toggled to 'Printer Friendly Mode' by the end user. |
| Page/Region is Read Only | Either the page or parent region, if appropriate, 'Read Only' condition evaluates to TRUE. |
| Page/Region is NOT Read Only | The page and parent region, if appropriate, are displayed normally. |

Table C-1 (Cont.) Available Conditions

| Condition | Description |
|--|---|
| User is authenticated (not public) | The end user is logged into the application. See Also: " Establishing User Identity Through Authentication " for information about authentication |
| User is the public user (user has not authenticated) | The end user is not logged into the application. |
| Inline Validation Errors Displayed | An inline validation error is displayed on the page, following a page submission. |
| Inline Validation Errors NOT displayed | An inline validation error is not displayed on the page. |
| Current Language = Value | The end user is currently running the application in the language you enter into the Value attribute. |
| Current Language != Value | The end user is not currently running the application in the language you enter into the Value attribute. |
| Current Language is contained in Value | The end user is currently running the application in one of the languages you enter into the Value attribute. |
| Current Language is NOT contained in Value | The end user is not currently running the application in one of the languages you enter into the Value attribute. |
| Never | The component is never rendered. Note: Changing a condition to Never will remove any existing condition. For temporarily disabling a component, Oracle recommends using the Build Option attribute. |

D

About Granted Privileges

Learn about grant privileges in Oracle Application Express.

- [About Granting Privileges to Database Users](#)
- [Understanding Privileges Granted to PUBLIC](#)

About Granting Privileges to Database Users

In Oracle Application Express, no privileges are granted to database users when creating a workspace on an existing schema. Unless direct grants exist, database users in Oracle Application Express will not be able to create tables or other database objects. To allow the creation of typical database objects, Oracle recommends the following list of directly granted privileges:

```
grant CREATE SESSION to [database username]
grant CREATE CLUSTER to [database username]
grant CREATE DIMENSION to [database username]
grant CREATE INDEXTYPE to [database username]
grant CREATE JOB to [database username]
grant CREATE MATERIALIZED VIEW to [database username]
grant CREATE OPERATOR to [database username]
grant CREATE PROCEDURE to [database username]
grant CREATE SEQUENCE to [database username]
grant CREATE SNAPSHOT to [database username]
grant CREATE SYNONYM to [database username]
grant CREATE TABLE to [database username]
grant CREATE TRIGGER to [database username]
grant CREATE TYPE to [database username]
grant CREATE VIEW to [database username]
```

Understanding Privileges Granted to PUBLIC

This section describes public synonyms that exist and for which the execute privilege is granted to PUBLIC for the packages, procedures, functions, tables, and views owned by the Oracle Application Express product schema, (for example, APEX_180200). To avoid conflicts with operation of the Application Express engine, Oracle recommends

not to use the following names for database objects in schemas associated with Oracle Application Express workspaces.

 **Tip:**

To avoid conflicts with operation of the Application Express engine, Oracle recommends not to use the following names for database objects in schemas associated with Oracle Application Express workspaces.

- Packages
- Procedures
- Functions
- Tables
- Views
- Sequences
- Types

Packages

Public synonyms exist and execute privilege is granted to PUBLIC for the following packages owned by the Oracle Application Express product schema, (for example, APEX_180200):

```
APEXWS  
  
APEX_MIG_PROJECTS_UPDATE  
  
APEX_UI_DEFAULT_UPDATE  
  
HTMLDB_CUSTOM_AUTH  
  
HTMLDB_ITEM  
  
HTMLDB_LANG  
  
HTMLDB_SITE_ADMIN_PRIVS  
  
WWV_FLOW_APPLICATION_INSTALL  
  
HTMLDB_UTIL  
  
WWV_FLOW  
  
WWV_FLOW_API  
  
WWV_FLOW_AUDIT  
  
WWV_FLOW_CSS_API  
  
WWV_FLOW_APPLICATION_INSTALL  
  
WWV_FLOW_AUDIT  
  
WWV_FLOW_COLLECTION
```

WWV_FLOW_CSS
WWV_FLOW_CSS_API
WWV_FLOW_CUSTOMIZE
WWV_FLOW_CUSTOM_AUTH
WWV_FLOW_CUSTOM_AUTH_LDAP
WWV_FLOW_CUSTOM_AUTH_STD
WWV_FLOW_DEBUG_MESSAGE
WWV_FLOW_EPG_INCLUDE_MODULES
WWV_FLOW_FILE_MGR
WWV_FLOW_FND_USER_API
WWV_FLOW_GLOBAL
WWV_FLOW_HINT
WWV_FLOW_HTML_API
WWV_FLOW_ID
WWV_FLOW_IMAGE_API
WWV_FLOW_ITEM
WWV_FLOW_ITEM_HELP
WWV_FLOW_JAVASCRIPT
WWV_FLOW_LANG
WWV_FLOW_LDAP
WWV_FLOW_LOOKUP_TABLES
WWV_FLOW_MAIL
WWV_FLOW_PLSQL_JOB
WWV_FLOW_PLUGIN
WWV_FLOW_PLUGIN_UTIL
WWV_FLOW_PREFERENCES
WWV_FLOW_RANDOM
WWV_FLOW_RENDER_SHORTCUT
WWV_FLOW_REST
WWV_FLOW_SVG
WWV_FLOW_USER_API
WWV_FLOW_UTILITIES
WWV_FLOW_WEBSERVICES_API

WWV_FLOW_WS_IMPORT_API

WWV_MIG_ACC_LOAD

WWV_RENDER_CALENDAR2

WWV_RENDER_CHART2

Execute privilege is granted to PUBLIC for the following packages owned by the Oracle Application Express product schema, (for example, APEX_180200):

WWV_CALCULATOR

WWV_FLOW_FLASH_CHART

WWV_FLOW_GENERATE_DDL

WWV_FLOW_IMAGE_GENERATOR

WWV_FLOW_TREE_GLOBAL_VARS

Procedures

Public synonyms exist and execute privilege is granted to PUBLIC for the following procedures owned by the Oracle Application Express product schema, (for example, APEX_180200):

APEX

APEX_ADMIN

DEVELOPMENT_SERVICE_HOME

DEVELOPMENT_SERVICE_HOME_LOGIN

DEVELOPMENT_SERVICE_SIGNUP

F

HTMLDB

HTMLDB_ADMIN

HTMLDB_LOGIN

P

WS

WWV_FLOW_INIT_HTTP_BUFFER

Z

Functions

Public synonyms exist and execute privilege is granted to PUBLIC for the following functions owned by the Oracle Application Express product schema, (for example, apex_180200):

NV

V

Execute privilege is granted to PUBLIC for the following function owned by the Oracle Application Express product schema, (for example, apex_180200):

WWV_POPUP_FILTER

Tables

Public synonyms exist and select privilege is granted to PUBLIC for the following tables owned by the Oracle Application Express product schema, (for example, APEX_180200):

WWV_FLOW_DUAL100

WWV_FLOW_TEMP_TABLE

WWV_FLOW_LOV_TEMP

Views

Public synonyms exist and select privilege is granted to PUBLIC for the following views owned by the Oracle Application Express product schema (for example, APEX_180200):

APEX_APPLICATIONS

APEX_APPLICATION_ALL_AUTH

APEX_APPLICATION_AUTH

APEX_APPLICATION_AUTHORIZATION

APEX_APPLICATION_BC_ENTRIES

APEX_APPLICATION_BREADCRUMBS

APEX_APPLICATION_BUILD_OPTIONS

APEX_APPLICATION_CACHING

APEX_APPLICATION_COMPUTATIONS

APEX_APPLICATION_GROUPS

APEX_APPLICATION_ITEMS

APEX_APPLICATION_LISTS

APEX_APPLICATION_LIST_ENTRIES

APEX_APPLICATION_LOVS

APEX_APPLICATION_LOV_ENTRIES

APEX_APPLICATION_NAV_BAR

APEX_APPLICATION_PAGES

APEX_APPLICATION_PAGE_BRANCHES

APEX_APPLICATION_PAGE_BUTTONS

APEX_APPLICATION_PAGE_CHARTS
APEX_APPLICATION_PAGE_CHART_S
APEX_APPLICATION_PAGE_CHART_A
APEX_APPLICATION_PAGE_COMP
APEX_APPLICATION_PAGE_DA
APEX_APPLICATION_PAGE_DA_ACTS
APEX_APPLICATION_PAGE_DB_ITEMS
APEX_APPLICATION_PAGE_FLASH5
APEX_APPLICATION_PAGE_FLASH5_S
APEX_APPLICATION_PAGE_GROUPS
APEX_APPLICATION_PAGE_IR
APEX_APPLICATION_PAGE_IR_CAT
APEX_APPLICATION_PAGE_IR_CGRPS
APEX_APPLICATION_PAGE_IR_COL
APEX_APPLICATION_PAGE_IR_COMP
APEX_APPLICATION_PAGE_IR_COND
APEX_APPLICATION_PAGE_IR_GRPBY
APEX_APPLICATION_PAGE_IR_PIVOT
APEX_APPLICATION_PAGE_IR_PVAGG
APEX_APPLICATION_PAGE_IR_PVSRT
APEX_APPLICATION_PAGE_IR_RPT
APEX_APPLICATION_PAGE_IR_SUB
APEX_APPLICATION_PAGE_ITEMS
APEX_APPLICATION_PAGE_MAP
APEX_APPLICATION_PAGE_PROC
APEX_APPLICATION_PAGE_REGIONS
APEX_APPLICATION_PAGE_RPT
APEX_APPLICATION_PAGE_RPT_COLS
APEX_APPLICATION_PAGE_TREES
APEX_APPLICATION_PAGE_VAL
APEX_APPLICATION_PARENT_TABS
APEX_APPLICATION_PROCESSES
APEX_APPLICATION_SETTINGS

APEX_APPLICATION_SHORTCUTS
APEX_APPLICATION_SUBSTITUTIONS
APEX_APPLICATION_SUPP_OBJECTS
APEX_APPLICATION_SUPP_OBJ_BOPT
APEX_APPLICATION_SUPP_OBJ_CHK
APEX_APPLICATION_SUPP_OBJ_SCR
APEX_APPLICATION_TABS
APEX_APPLICATION_TEMPLATES
APEX_APPLICATION_TEMP_BC
APEX_APPLICATION_TEMP_BUTTON
APEX_APPLICATION_TEMP_CALENDAR
APEX_APPLICATION_TEMP_LABEL
APEX_APPLICATION_TEMP_LIST
APEX_APPLICATION_TEMP_PAGE
APEX_APPLICATION_TEMP_POPUPLOV
APEX_APPLICATION_TEMP_REGION
APEX_APPLICATION_TEMP_REPORT
APEX_APPLICATION_THEMES
APEX_APPLICATION_TRANSLATIONS
APEX_APPLICATION_TRANS_MAP
APEX_APPLICATION_TREES
APEX_APPLICATION_WEB_SERVICES
APEX_APPL_ACL_ROLES
APEX_APPL_ACL_USER_ROLES
APEX_APPL_ACL_USERS
APEX_APPL_PAGE_IG_COLUMNS
APEX_APPL_PAGE_IG_COL_GROUPS
APEX_APPL_PAGE_IGS
APEX_APPL_PAGE_IG_RPTS
APEX_APPL_PAGE_IG_RPT_VIEWS
APEX_APPL_PAGE_IG_RPT_COMPUTES
APEX_APPL_PAGE_IG_RPT_FILTERS
APEX_APPL_PAGE_IG_RPT_COLUMNS

APEX_APPL_PAGE_IG_RPT_HIGHLTS
APEX_APPL_PAGE_IG_RPT_AGGS
APEX_APPL_PAGE_IG_RPT_CHT_COLS
APEX_APPL_PLUGINS
APEX_APPL_PLUGIN_ATTRIBUTES
APEX_APPL_PLUGIN_ATTR_VALUES
APEX_APPL_PLUGIN_EVENTS
APEX_APPL_PLUGIN_FILES
APEX_DEBUG_MESSAGES
APEX_DEVELOPER_ACTIVITY_LOG
APEX_DICTIONARY
APEX_MIGRATION_ACC_FORMS
APEX_MIGRATION_ACC_PROJECTS
APEX_MIGRATION_ACC_QUERIES
APEX_MIGRATION_ACC_RPTS
APEX_MIGRATION_ACC_TABLES
APEX_MIGRATION_PROJECTS
APEX_MIGRATION_REV_FORMS
APEX_MIGRATION_REV_QUERIES
APEX_MIGRATION_REV_RPTS
APEX_MIGRATION_REV_TABLES
APEX_PKG_APP_ACTIVITY
APEX_PKG_APP_AUTHENTICATIONS
APEX_PKG_APPS
APEX_RELEASE
APEX_STANDARD_CONDITIONS
APEX_TEAM_BUGS
APEX_TEAM_FEEDBACK
APEX_TEAM_MILESTONES
APEX_TEAM_TODOS
APEX_THEMES
APEX_UI_DEFAULTS_ATTR_DICT
APEX_UI_DEFAULTS_COLUMNS

APEX_UI_DEFAULTS_GROUPS
APEX_UI_DEFAULTS_LOV_DATA
APEX_UI_DEFAULTS_TABLES
APEX_WORKSPACES
APEX_WORKSPACE_ACCESS_LOG
APEX_WORKSPACE_ACTIVITY_LOG
APEX_WORKSPACE_APEX_USERS
APEX_WORKSPACE_CLICKS
APEX_WORKSPACE_DEVELOPERS
APEX_WORKSPACE_FILES
APEX_WORKSPACE_LOG_SUMMARY
APEX_WORKSPACE_LOG_SUMMARY_USR
APEX_WORKSPACE_SCHEMAS
APEX_WORKSPACE_SESSIONS
APEX_WORKSPACE_SQL_SCRIPTS
APEX_WS_APPLICATIONS
APEX_WS_APP_PAGES
APEX_WS_DATA_GRID
APEX_WS_DATA_GRID_COL
WWV_FLOW_CLICKTHRU_LOG
WWV_FLOW_COLLECTIONS
WWV_FLOW_FEEDBACK_TYPES
WWV_FLOW_GROUP_USERS
WWV_FLOW_LISTS_OF_VALUES\$_V
WWV_FLOW_LIST_OF_VALUES_DATA_V
WWV_FLOW_MONTHS_MON
WWV_FLOW_MONTHS_MONTH
WWV_FLOW_PLSQL_JOBS
WWV_FLOW_USERS
WWV_FLOW_USER_ACCESS_LOG
WWV_FLOW_USER_ACTIVITY_LOG
WWV_FLOW_USER_MAIL_LOG
WWV_FLOW_YEARS

Additionally, insert is granted on the following:

WWV_FLOW_FILES

Additionally, the following views have select privilege granted to PUBLIC, but no synonym:

WWV_FLOW_HOURS_12

WWV_FLOW_HOURS_24

WWV_FLOW_MINUTES

WWV_FLOW_MINUTES_5

Sequences

The following sequences have public synonyms and select privilege granted to public:

WWV_FLOW_SESSION_SEQ

WWV_SEQ

Types

The following types have execute granted to public but no synonyms:

WWV_FLOW_TREE_ENTRY

WWV_FLOW_TREE_SUBS

Index

A

- Accept Page, [2-6](#)
- access control, [20-52](#)
 - configuring, [20-45](#)
 - creating, [20-43](#)
 - creating with Create Application Wizard, [6-7](#)
 - creating with Create Page Wizard, [7-6](#)
- Access Control
 - about exporting, [20-46](#)
 - authorization schemes, [20-44](#)
- access control list, [20-42](#)
 - adding multiple users, [20-49](#)
 - adding users, [20-49](#)
 - adding with Create Application Wizard, [20-47](#)
 - adding with Create Page Wizard, [20-48](#)
 - configuring, [20-50](#)
 - deleting users, [20-49](#)
 - editing users, [20-49](#)
- access control, authorization schemes, [20-50](#)
- Account menu, [1-12](#)
- activity log, [22-9](#)
- Activity Monitoring
 - creating with Create Page Wizard, [7-7](#)
- Activity Reporting
 - creating with Create Application Wizard, [6-7](#)
- activity reports, creating custom, [22-9](#)
- Administration menu, [1-10](#)
- Administration Menu
 - Administration, [1-10](#)
 - Change My Password, [1-10](#)
 - Dashboards, [1-10](#)
 - Manage Service, [1-10](#)
 - Manage Users and Groups, [1-10](#)
 - Monitory Activity, [1-10](#)
- Allow Feedback attribute, [6-30](#)
- Allows URLs Created After attribute, [6-41](#)
- AnyChart, [11-4](#)
- AnyCo IT Department
 - about, [3-22](#)
 - installing, [3-22](#)
 - navigating, [3-23](#)
 - running, [3-22](#)
- AnyGantt, [11-4](#)
- APEX_ACTIVITY_LOG view, [22-9](#)
- APEX_MAIL, [19-3](#)
- App Administration
 - making service requests, [3-38](#)
 - Request Storage, [3-38](#)
 - Request Termination, [3-38](#)
 - setting workspace preferences, [3-39](#)
- App Builder
 - accessing, [1-14](#), [1-22](#)
 - home page, [1-15](#)
 - page, [2-2](#)
- App Builder home page, [1-14](#), [1-15](#), [1-22](#)
 - Create button, [1-18](#)
 - Migrations, [1-19](#)
 - navigation bar, [1-16](#)
 - Recent, [1-19](#)
 - Reset button, [1-18](#)
- App Builder Tasks
 - Recent list, [1-19](#)
- app gallery
 - managing, [3-2](#)
- App Gallery
 - productivity application, [3-1](#)
 - sample application, [3-1](#)
- App Gallery page, [3-3](#)
 - accessing, [3-3](#)
- APP_AJAX_X01, [2-30](#)
- APP_AJAX_X10, [2-30](#)
- application
 - adding content from Web sites, [13-14](#)
 - adding Data Load Wizard, [18-1](#)
 - adding data loading, [18-1](#)
 - attaching an authorization scheme, [20-77](#)
 - changing authentication scheme, [20-70](#), [20-71](#)
 - controlling navigation using branches, [15-23](#)
 - copying, [1-27](#)
 - creating, [2-2](#)
 - debugging, [22-1](#), [23-1](#)
 - defining primary language, [21-7](#)
 - deleting, [1-27](#)
 - deploying, [24-4](#), [24-6](#), [24-27](#)
 - exporting, [24-6](#), [24-27](#)
 - grouping, [6-68](#)
 - hiding features, [24-22](#)
 - importing export file, [24-38](#)

- application (*continued*)
 - importing from SQL*Plus, [24-46](#)
 - language preference, [21-3](#)
 - resource use, [23-3](#)
 - translatable components, [21-4](#)
 - translating, [21-2](#)
 - translating multibyte languages, [21-8](#)
 - translation rendering, [21-4](#)
 - upgrading, [6-74](#)
- application access control
 - about, [20-52](#)
 - adding user role assignments, [20-54](#)
 - creating roles, [20-53](#)
 - deleting roles, [20-54](#)
 - deleting user role assignments, [20-55](#)
 - editing roles, [20-54](#)
 - editing user role assignments, [20-55](#)
- Application Alias attribute, [6-28](#)
- application attributes
 - Build Options, [6-35](#)
 - Database Session, [6-43](#)
 - editing, [6-26](#)
 - Error Handling, [6-33](#)
 - exporting, [24-30](#)
 - Substitutions, [6-35](#)
- application blueprint
 - about, [6-19](#)
 - blueprint
 - using, [6-19](#)
 - loading, [6-20](#)
 - using, [6-19](#)
 - viewing, [6-20](#)
- application computations
 - about, [16-22](#)
 - creating, [16-23](#)
 - executing On New Instance, [16-17](#), [16-23](#)
 - history report, [16-24](#)
- Application Dashboard report, [22-5](#)
- Application Date Format attribute, [6-46](#)
- Application Date Time Format attribute, [6-46](#)
- application definition, editing, [6-26](#)
- Application Express Accounts authentication
 - setting up, [20-60](#), [20-69](#)
- Application Express Views, [6-76](#)
- application groups, [6-68](#)
- Application Groups attribute, [6-29](#)
- application Help, creating, [11-41](#)
- Application home page
 - adding new page, [7-2](#)
 - Export/Import icon, [1-24](#)
 - Export/Install icon, [3-20](#)
 - Run Application button, [1-24](#)
 - Shared Components icon, [1-24](#), [3-20](#)
 - Supporting Objects icon, [1-24](#), [3-20](#)
 - Utilities icon, [1-24](#), [3-20](#)
- Application home Task list
 - Copy this Application, [1-27](#)
 - Delete this Application, [1-27](#)
- application item
 - about, [14-19](#), [17-37](#)
 - creating, [17-37](#)
 - creating a global variable, [17-40](#)
 - editing attributes, [17-39](#)
 - history, [17-39](#)
- application language, [6-44](#)
- Application Language Derived From attribute, [6-46](#), [21-7](#)
- application layout
 - print preview mode, [13-2](#)
 - shortcuts, [17-19](#)
- Application Primary Language attribute, [6-45](#), [21-7](#)
- application processes
 - creating, [16-19](#)
 - creating error message, [16-20](#)
 - editing attributes, [16-20](#)
 - example, [16-19](#)
 - On Demand, [16-17](#)
 - reports, [16-21](#)
 - understanding, [16-16](#)
- application settings, [17-9](#)
 - creating, [17-10](#)
 - deleting, [17-10](#)
 - editing, [17-10](#)
- Application Timestamp Format, [6-46](#)
- Application Timestamp Time Zone Format, [6-47](#)
- application utilities
 - accessing, [6-72](#)
 - Advisor, [6-71](#)
 - Application Dashboard, [6-71](#)
 - Application Express Views, [6-71](#)
 - Attribute Dictionary, [6-71](#)
 - Change History, [6-71](#)
 - Database Object Dependencies, [6-71](#)
 - Debug Messages, [6-71](#)
 - Export Repository, [6-71](#)
 - Recently Updated Pages, [6-71](#)
 - Upgrade Application, [6-71](#)
 - using, [6-70](#)
- Apps Administration
 - managing saved interactive reports, [3-40](#)
 - managing subscriptions, [3-41](#)
- architecture, [1-3](#)
- asynchronous updates
 - charts, [25-10](#)
 - maps, [11-28](#)
- Attribute Dictionary, [16-24](#)
 - about, [16-25](#)
 - accessing, [16-25](#)

- attributes
 - viewing field-level help, [1-31](#)
- authentication, [2-9](#), [20-56](#)
 - Authentication Schemes Repository, [20-70](#)
 - changing, [20-70](#), [20-71](#)
 - creating an authentication scheme, [20-58](#)
 - managing existing schemes, [20-70](#)
 - preconfigured authentication schemes, [20-59](#)
 - reports, [20-72](#)
 - security attributes, [6-37](#)
 - understanding, [20-56](#)
 - viewing current scheme, [20-70](#)
- authentication report
 - Authentication Scheme History, [20-72](#)
 - Authentication Scheme Subscription, [20-72](#)
- Authentication Scheme attribute, [6-37](#)
- Authentication Scheme History report, [20-72](#)
- Authentication Scheme Subscription report, [20-72](#)
- authentication schemes
 - Application Express Accounts, [20-60](#)
 - cookie user accounts, [20-60](#)
 - Custom authentication, [20-61](#)
 - DAD Credentials, [20-66](#)
 - Database Account Credentials, [20-62](#)
 - HTTP Header Variable, [20-63](#)
 - LDAP Directory Verification, [20-64](#)
 - Open Door Credentials, [20-67](#)
 - Single Sign-On verification, [20-67](#)
 - Social Sign-In, [20-68](#)
- authorization, [2-9](#)
- Authorization attributes, [6-38](#)
- authorization scheme
 - attaching to application, [20-77](#)
 - attaching to component, [20-78](#)
 - attaching to control, [20-78](#)
 - attaching to page, [20-77](#)
 - By Component report, [20-78](#)
 - History report, [20-78](#)
 - of an application, [6-38](#)
 - reports, [20-78](#)
 - specifying, [6-38](#)
 - Subscription report, [20-78](#)
 - Utilization report, [20-78](#)
- Authorization Scheme attribute, [6-38](#)
- Authorization Scheme History report, [20-78](#)
- Authorization Scheme Subscription report, [20-78](#)
- Authorization Scheme Utilization report, [20-78](#)
- authorization schemes, [20-72](#)
 - attaching to application, [20-77](#)
 - attaching to component, [20-78](#)
 - attaching to control, [20-78](#)
 - attaching to page, [20-77](#)
 - creating, [20-73](#)
 - for controlling access, [20-50](#)

- Automatic CSV Encoding attribute, [6-48](#)
- Automatic Time Zone, [6-47](#)

B

- bind variables
 - syntax, [2-21](#)
 - using in regions, [2-21](#)
 - using in regions based on PL/SQL, [2-22](#)
- blank page
 - creating with Create App Wizard, [6-5](#)
- BLOB support in forms
 - about, [10-23](#)
 - displaying the BLOB, [10-24](#)
 - downloading files, [10-23](#)
 - removing the image reference, [10-24](#)
 - uploading files, [10-23](#)
- BLOB support in reports
 - about, [9-108](#)
 - displaying the BLOB, [9-109](#)
 - providing download link, [9-108](#)
- BLOBs
 - APEX_UTIL.GET_BLOB_FILE_SRC, [9-110](#)
 - download support, [10-23](#)
 - populating, [10-23](#)
 - support, [9-107](#)
 - support in forms, [10-22](#)
 - working with procedurally, [9-110](#)
- Bookmark Hash Function attribute, [6-41](#)
- branch
 - about, [16-16](#)
 - creating, [15-23](#)
 - page submission, [8-30](#)
 - pre-rendering, [8-30](#)
 - When Button Pressed attribute, [2-43](#)
- BRANCH_TO_PAGE_ACCEPT, [2-44](#)
- branching
 - making conditional, [15-25](#)
 - using buttons, [14-38](#)
- breadcrumb reports, [15-16](#)
 - Breadcrumb Exceptions, [15-16](#)
 - Breadcrumb Hierarchy, [15-16](#)
 - Breadcrumb History, [15-16](#)
 - Breadcrumb Utilization, [15-16](#)
 - Grid Edit Breadcrumb, [15-16](#)
- breadcrumb template, [12-46](#)
 - Definition, [12-48](#)
 - Name, [12-48](#)
 - Substitution Strings, [12-48](#)
- breadcrumb template substitution strings
 - #LINK#, [12-49](#)
 - #LONG_NAME#, [12-49](#)
 - #NAME#, [12-49](#)
 - NAME_ESC_SC#, [12-49](#)

breadcrumbs
 about, [15-9](#)
 about creating manually, [15-10](#)
 adding a breadcrumb region, [15-12](#)
 adding entries, [15-13](#)
 adding to a page, [15-12](#)
 creating, [15-8](#)
 creating as shared components, [15-11](#)
 creating dynamic, [15-14](#)
 creating with Breadcrumb Region Wizard, [15-11](#)
 deleting unused, [15-16](#)
 editing, [15-15](#)
 Exceptions report, [15-16](#)
 Grid Edit report, [15-16](#)
 Hierarchy report, [15-16](#)
 History report, [15-16](#)
 navigations, [12-47](#)
 reparenting entries, [15-15](#)
 Utilization report, [15-16](#)
 Browse button, [A-4](#)
 bug reports
 By Day, [4-23](#)
 By Day page, [4-23](#)
 By Developer, [4-23](#)
 Calendar page, [4-22](#)
 bugs
 attaching files, [4-21](#)
 Bugs page, [4-20](#)
 creating, [4-20](#)
 editing, [4-20](#)
 managing, [4-19](#)
 viewing, [4-20](#)
 viewing reports, [4-22](#)
 build options, [24-16](#)
 about, [24-17](#)
 changing status, [24-18](#)
 creating, [24-18](#)
 deleting, [24-19](#), [24-20](#)
 excluding, [24-18](#)
 exporting, [24-22](#)
 exporting status, [24-22](#)
 history, [24-21](#)
 including, [24-18](#)
 removing associated components, [24-19](#), [24-20](#)
 reports, [24-21](#)
 Build Options attribute, [6-35](#)
 Build Status attribute, [6-32](#)
 built-in substitution strings, [2-26](#)
 Bulk Edit Item Help report
 accessing, [11-44](#)
 button
 branching, [14-38](#)
 creating, [14-33](#)

button (*continued*)
 displaying conditionally, [14-35](#)
 editing, [14-34](#), [14-35](#)
 managing, [14-32](#)
 names, [14-37](#)
 template, [12-49](#)
 Warn on Unsaved Changes, [14-35](#)
 button template
 Definition, [12-50](#)
 Name, [12-50](#)
 Substitution Strings, [12-50](#)
 button template substitution strings
 ##ICON_CSS_CLASSES##, [12-51](#)
 #BUTTON_ATTRIBUTES#, [12-51](#)
 #BUTTON_CSS_CLASSES#, [12-51](#)
 #BUTTON_ID#, [12-51](#)
 #LABEL#, [12-51](#)
 #LINK#, [12-51](#)

C

Cache attribute, [6-42](#)
 caching regions, [13-9](#)
 calendar
 accessing attributes, [11-14](#), [25-20](#)
 add initialization JavaScript code, [11-17](#)
 adding, [11-9](#), [11-10](#)
 adding to new page, [11-9](#), [11-10](#)
 creating, [11-6](#)
 creating in Page Designer, [11-12](#)
 creating legacy calendar, [25-20](#)
 creating with Create App Wizard, [6-6](#)
 creating with Create Page Wizard, [7-6](#)
 creating with REST Enabled SQL, [18-23](#)
 editing attributes, [11-14](#)
 enabling drag and drop, [11-18](#)
 including add and edit functions, [11-15](#)
 integrating with Google calendar, [17-16](#)
 supported substitution strings, [25-21](#)
 viewing sample, [11-8](#)
 calendar attributes
 accessing, [11-14](#), [25-20](#)
 cards
 creating with Create App Wizard, [6-6](#)
 cascading style sheets
 about, [12-32](#)
 referencing in page template, [12-81](#)
 uploading, [12-81](#)
 using custom, [12-81](#)
 Change History report, [6-75](#)
 Character Value Comparison, [6-47](#)
 Character Value Comparison Behavior, [6-47](#)
 chart
 creating with Create Page Wizard, [7-5](#)

- chart synchronous update
 - creating refresh dynamic action, [11-39](#)
- chart type
 - switching in page designer, [11-37](#)
- charts
 - adding to a page, [11-35](#)
 - adding to new page, [11-31](#), [11-32](#)
 - altering display, [11-37](#)
 - AnyChart, [11-4](#)
 - AnyGantt, [11-4](#)
 - asynchronous updates, [25-10](#)
 - automatic refresh, [11-38](#)
 - automatic refresh attribute
 - enabling, [11-38](#)
 - creating, [11-29](#)
 - creating with Create App Wizard, [6-6](#)
 - creating with REST Enabled SQL, [18-20](#)
 - custom JavaScript, [11-40](#)
 - dynamic action, [11-38](#)
 - editing attributes, [11-37](#)
 - legacy, [25-1](#)
 - managing in Page Designer, [11-36](#)
 - monitoring information, [25-10](#)
 - switching to Flash, [25-9](#)
 - upgrading AnyChart to JET Chart, [11-4](#)
- check box, [A-2](#)
- classic calendar template
 - comments, [12-52](#)
 - Daily Calendar, [12-53](#)
 - Monthly Calendar, [12-53](#)
 - Name, [12-52](#)
 - Template Subscription, [12-52](#)
 - Weekly Calendar, [12-53](#)
- classic navigation bar, [15-17](#)
 - about, [15-18](#)
 - copying entry, [15-21](#)
 - creating entry, [15-18](#)
 - creating for feedback, [15-20](#)
 - editing entry, [15-21](#)
 - editing multiple entries, [15-22](#)
 - History report, [15-22](#)
 - reports, [15-22](#)
 - Subscription report, [15-22](#)
- classic report
 - adding download link, [9-102](#)
 - changing pagination, [9-92](#)
 - controlling column breaks, [9-94](#)
 - editing attributes, [9-91](#)
 - support for file export, [9-93](#)
- classic report regions
 - configuring print attributes, [9-105](#)
 - printing methods, [9-104](#)
- classic reports
 - about, [9-5](#)
 - controlling column display, [9-101](#)
- classic reports (*continued*)
 - creating a column link, [9-98](#)
 - creating updatable columns, [9-99](#)
 - creating with Create Application Wizard, [9-7](#)
 - defining a column as a list of values, [9-100](#)
 - editing column layout, [9-96](#)
 - managing Attributes, [9-90](#)
 - managing column attributes, [9-95](#)
 - managing region attributes, [9-89](#)
 - sorting columns, [9-97](#)
- Cleanup PL/SQL Code, [6-43](#)
- collections, [18-8](#)
- color picker, [A-2](#)
 - configure display style, [17-15](#), [17-16](#)
- columns templates, using JavaScript, [12-80](#)
- commit points, [2-7](#)
- Compatibility Mode attribute, [6-30](#)
- Component Settings
 - adding sliders, [17-17](#)
 - Color Picker, [17-15](#), [17-16](#)
 - integrating with Google calendar, [17-16](#)
 - interactive report Action menu, [17-17](#)
 - managing, [17-15](#)
 - switch defaults, [17-18](#)
 - viewing History report, [17-19](#)
- Component View tab
 - adding buttons, [8-28](#)
 - adding items, [8-28](#)
 - adding regions, [8-28](#)
- components
 - attaching an authorization scheme, [20-78](#)
 - controlling access to, [2-9](#)
 - displaying on all pages, [13-3](#)
 - displaying on every page, [13-2](#)
 - managing availability, [20-16](#)
 - translating messages, [21-16](#), [21-17](#), [21-38](#)
- condition type, [2-8](#)
- conditional
 - processing, [2-7](#)
 - rendering, [2-7](#)
- conditions
 - list of, [C-1](#)
 - using, [2-7](#)
 - using with buttons, [14-35](#)
- Configuration Options
 - creating with Create Application Wizard, [6-7](#)
 - creating with Create Page Wizard, [7-7](#)
- configuration, controlling, [24-16](#)
- controls, controlling access to, [2-9](#)
- Create App Wizard, Advanced Settings
 - Description, [6-10](#)
 - Security, [6-10](#)
 - Settings, [6-10](#)
 - User Interface Attributes, [6-10](#)

- Create Application Wizard, [2-2](#)
 - about, [6-2](#), [6-5](#)
 - About Page
 - creating with Create Application Wizard, [6-7](#)
 - access control, [6-7](#)
 - Activity Monitoring, [6-7](#)
 - adding blank page, [6-5](#)
 - adding calendar, [6-6](#)
 - adding cards, [6-6](#)
 - adding chart, [6-6](#)
 - adding classic report, [9-10](#)
 - adding dashboard, [6-6](#)
 - adding form, [6-6](#)
 - adding interactive grid, [6-6](#), [9-10](#)
 - adding interactive report, [9-10](#)
 - adding master detail, [6-6](#)
 - adding multiple reports, [6-7](#)
 - adding report, [6-6](#)
 - adding timeline, [6-6](#)
 - adding wizard, [6-6](#)
 - Configuration Options, [6-7](#)
 - creating Feedback, [6-8](#)
 - creating forms, [10-8](#)
 - creating master detail forms, [10-15](#)
 - creating reports, [9-6](#), [9-8](#)
 - page types, [6-5](#)
 - report options, [9-10](#)
 - Settings, [6-8](#)
 - Theme Style Selection, [6-8](#)
- Create Application Wizard, Settings
 - Application ID, [6-8](#)
 - Authentication, [6-9](#)
 - Schema, [6-9](#)
 - User Interface Defaults, [6-11](#)
- Create Branch Wizard, [15-23](#)
- Create Navigation Bar Entry Wizard, [11-45](#), [15-20](#)
- Create Page Computation Wizard, [16-2](#)
- Create Page Wizard, [7-1](#)
 - About Page, [7-6](#)
 - About this Application
 - creating with Create Page Wizard, [7-6](#)
 - access control, [7-6](#)
 - Activity Monitoring, [7-7](#)
 - adding Drill Down master detail, [10-7](#)
 - adding interactive grid, [10-6](#)
 - adding Side by Side master detail, [10-7](#)
 - adding Stacked master detail, [10-7](#)
 - blank page
 - creating with Create Page Wizard, [7-4](#), [7-8](#)
 - Configuration Options, [7-7](#)
 - creating blank page, [7-4](#), [7-8](#)
 - creating calendar, [7-6](#), [7-9](#)
- Create Page Wizard (*continued*)
 - creating chart, [7-5](#), [7-9](#)
 - creating classic report, [9-15](#)
 - creating column toggle report, [9-15](#), [9-16](#)
 - creating Dashboard, [7-5](#)
 - creating Data Load Wizard
 - Data Load Wizard, [18-2](#)
 - creating data loading wizard, [7-9](#)
 - creating Data Loading wizard, [7-6](#), [7-9](#)
 - creating Feedback, [7-7](#)
 - creating form, [7-5](#), [7-9](#)
 - creating form on a procedure, [10-8](#)
 - creating form on a SQL query, [10-7](#)
 - creating form on procedure, [10-7](#)
 - creating form on SQL Query, [10-8](#)
 - creating form on table, [10-6](#)
 - creating form on Web service, [10-7](#)
 - creating forms, [10-4](#)
 - creating interactive grid, [9-15](#)
 - creating interactive report, [9-14](#)
 - creating Legacy page, [7-6](#), [7-9](#)
 - creating list view, [9-15](#), [9-16](#)
 - creating Login Page, [7-7](#)
 - creating master detail, [7-5](#)
 - creating plug-ins, [7-5](#), [7-9](#)
 - creating reflow report, [9-16](#)
 - creating report, [7-5](#), [7-9](#)
 - creating report and form on table, [9-15](#)
 - creating report and form on Web Service, [10-8](#)
 - creating report on Web service, [9-16](#)
 - creating report with a list view on table, [10-6](#), [10-8](#)
 - creating report with form on table, [10-6](#)
 - creating reports, [9-11](#), [9-12](#)
 - creating tree, [7-6](#), [7-9](#)
 - creating wizard, [7-6](#), [7-9](#)
 - Dashboard
 - creating with Create Page Wizard, [7-5](#)
 - Data Load Wizard
 - about pages created, [18-2](#)
 - Data Load Definition, [18-2](#)
 - Email Reporting, [7-7](#)
 - Form and Report on Web Service, [10-7](#)
 - form on table, [10-8](#)
 - form on Web Service, [10-8](#)
 - form options, [10-6](#)
 - Job Reporting, [7-7](#)
 - report options, [9-14](#)
 - Theme Style Selection, [7-8](#)
- Create Region Wizard
 - creating breadcrumb, [15-11](#)
 - creating forms, [10-10](#)
- credentials
 - about, [18-37](#)

- credentials (*continued*)
 - deleting, [18-38](#)
 - exporting, [18-37](#)
 - managing, [18-37](#)
 - cross page utilities
 - accessing, [6-77](#)
 - Delete Multiple Pages, [6-77](#)
 - Grid Edit of all Pages, [6-77](#)
 - Page Attributes, [6-77](#)
 - Page Locks, [6-77](#)
 - cross site-scripting
 - about, [20-24](#)
 - about safe item types, [20-25](#)
 - dynamic output, [20-26](#)
 - escaping values, [20-25](#)
 - HTML region, [20-24](#)
 - protecting dynamic output, [20-26](#)
 - protecting form items, [20-28](#)
 - protecting report regions, [20-27](#)
 - report regions, [20-27](#)
 - restricting characters on items, [20-28](#)
 - static display areas, [20-24](#)
 - using safe item types, [20-26](#)
 - CSS Finder, [6-63](#)
 - custom application
 - about, [24-8](#)
 - about Supporting Objects page, [24-9](#)
 - adding access control list, [24-12](#)
 - build options, [24-10](#)
 - creating supporting objects, [24-9](#)
 - creating with Supporting Objects utility, [24-7](#)
 - defining an upgrade script, [24-14](#)
 - deinstallation scripts, [24-12](#)
 - deinstalling, [24-15](#)
 - deleting installation options, [24-13](#)
 - deleting messages, [24-13](#)
 - deleting supporting object scripts, [24-13](#)
 - installation scripts, [24-10](#)
 - installing supporting objects, [24-13](#)
 - messages, [24-10](#), [24-12](#)
 - prerequisites, [24-10](#)
 - substitutions strings, [24-10](#)
 - testing installation scripts, [24-13](#)
 - upgrade messages, [24-11](#)
 - upgrade scripts, [24-11](#)
 - upgrading, [24-15](#)
 - validations, [24-10](#)
 - viewing an install summary, [24-16](#)
 - Custom Authentication
 - setting up, [20-61](#)
- ## D
-
- DAD Credentials Verification
 - setting up, [20-66](#)
 - Dashboard
 - creating with Create App Wizard, [6-6](#)
 - data
 - accessing with database links, [18-59](#)
 - accessing with REST Enabled SQL Reference, [18-9](#)
 - Data Load Wizard
 - about creating, [18-2](#)
 - adding to application, [18-1](#)
 - creating, [18-4](#)
 - creating with Create Page Wizard, [18-2](#)
 - re-creating pages, [18-6](#)
 - data loading
 - adding, [18-1](#)
 - Data Loading
 - creating with Create Page Wizard, [7-6](#), [7-9](#)
 - Database Account Credentials authentication
 - setting up, [20-63](#)
 - database application
 - about, [2-2](#)
 - adding a page, [7-1](#)
 - adding components, [7-1](#)
 - adding pages, [7-1](#)
 - assigning to groups, [6-69](#)
 - based on queries, [6-12](#)
 - based on spreadsheet, [6-13](#), [6-14](#)
 - copying, [6-21](#)
 - copying from Application home page, [6-22](#)
 - copying with Create Application Wizard, [6-21](#)
 - creating groups, [6-68](#)
 - creating on tables, [6-12](#)
 - deleting, [6-25](#)
 - groups, [6-68](#)
 - loading application blueprint, [6-20](#)
 - loading from spreadsheet, [6-17](#)
 - loading from spreadsheet file, [6-14](#)
 - removing from groups, [6-69](#)
 - running, [2-3](#)
 - using application blueprint, [6-19](#)
 - viewing application blueprint, [6-19](#), [6-20](#)
 - database application groups
 - assigning pages, [6-69](#)
 - creating, [6-68](#)
 - removing, [6-69](#)
 - removing pages, [6-69](#)
 - database application page
 - adding, [7-1](#)
 - assigning to groups, [7-18](#)
 - copying in Page Designer, [8-39](#)
 - creating from Application home, [7-2](#)
 - deleting, [8-40](#)
 - deleting in Page Designer, [8-40](#)
 - deleting multiple, [8-41](#)
 - deleting range of, [8-41](#)
 - determining lock status, [7-19](#)

- database application page (*continued*)
 - grouping, [7-16](#)
 - locking, [7-19](#), [7-20](#)
 - managing, [7-1](#)
 - preventing conflicts, [7-19](#)
 - removing from groups, [7-18](#)
 - running, [2-3](#)
 - unlocking, [7-19](#), [7-21](#)
- database application page groups, [7-17](#)
 - assigning pages, [7-18](#)
 - creating, [7-17](#)
 - editing group definition, [7-18](#)
 - removing, [7-19](#)
 - removing pages, [7-18](#)
 - viewing, [7-17](#)
- database link, [18-59](#)
 - about, [18-60](#)
 - creating, [18-60](#)
 - viewing existing database links, [18-60](#)
- Database Object Dependencies report, [6-75](#)
- Database Session attribute, [6-43](#)
- database users
 - granted privileges, [D-1](#)
 - granting privileges, [D-1](#)
- date picker, [A-3](#)
- DBMS_APPLICATION_INFO, [23-3](#)
- debugging, [22-1](#), [23-1](#)
 - checking for errors, [23-1](#)
 - enabling and disabling, [23-5](#)
 - isolating a problem, [23-9](#)
 - running with application, [23-6](#)
 - SQL queries, [22-3](#)
 - SQL tracing, [23-4](#)
 - using debug mode, [23-4](#)
 - using f?p to access, [23-7](#)
 - viewing reports at runtime, [23-7](#)
 - viewing reports from Component View, [23-8](#)
 - viewing reports in development mode, [23-6](#)
- Debugging attribute, [6-29](#)
- deep linking, [20-57](#)
- Deep Linking attribute, [6-39](#)
- Define Authorization Scheme, [6-38](#)
- defining, [18-42](#)
- deleting export files, [24-46](#)
- demonstration application
 - about Sample Database Application, [3-8](#)
- deployment, [24-6](#)
 - creating supporting objects, [24-9](#)
 - exporting a page, [24-32](#)
 - exporting a Websheet, [24-33](#)
 - exporting application definition, [24-27](#)
 - exporting feedback, [24-37](#)
 - exporting script files, [24-38](#)
 - exporting static files, [24-34](#)
 - exporting themes, [24-34](#)
- deployment (*continued*)
 - exporting User Interface Defaults, [24-35](#)
 - exporting workspaces, [24-31](#)
 - importing application from SQL*Plus, [24-46](#)
 - importing export files, [24-38](#)
 - importing files, [24-38](#)
 - installing application export, [24-44](#)
 - installing files, [24-45](#)
 - managing database objects, [24-6](#)
 - of an application, [24-4](#)
 - options, [24-5](#)
 - publishing the URL, [24-23](#)
 - publishing the Websheet application URL, [24-24](#)
 - understanding the process, [24-4](#)
- detail view
 - interactive grids, [9-35](#)
- developer comments
 - adding in Page Designer, [6-59](#)
 - adding to an application, [6-60](#)
 - adding to application, [6-60](#)
 - deleting, [6-61](#)
 - editing, [6-60](#)
 - viewing, [6-60](#)
 - viewing reports, [6-62](#)
- Developer Comments calendar, [6-62](#)
- Developer Navigation tools
 - about, [1-7](#)
 - Account menu, [1-12](#)
 - Administration menu, [1-10](#)
 - Help menu, [1-11](#)
- development process
 - managing, [4-1](#)
- development process, managing, [4-1](#)
- dialog
 - configuring attributes in Page Designer, [8-35](#)
- dialog page template, [7-11](#)
- dialog page template attributes, [12-67](#)
- dialog pages, [7-10](#)
 - About branching, [7-14](#)
 - about supported, [7-10](#)
 - creating a dialog page template, [7-11](#)
 - defining, [7-12](#)
 - generating a Cancel dynamic action, [7-13](#)
 - generating a close dialog page process, [7-13](#)
 - handling refresh actions, [7-14](#)
 - Modal Dialog, [7-10](#)
 - non-modal dialog, [7-10](#)
 - required template attributes, [12-67](#)
 - using dialog page template, [7-11](#)
- DML
 - about locking, [18-61](#)
 - APEX_DML_LOCK_WAIT_TIME, [18-61](#)
 - FSP_DML_LOCK_ROW, [18-61](#)

- doc library
 - downloading, [1-31](#)
- Document Direction attribute, [6-46](#)
- documentation
 - about, [1-29](#)
 - accessing, [1-29](#)
- documents
 - downloading, [1-31](#)
- downloading
 - doc library, [1-31](#)
- dynamic action
 - calendars, [11-19](#)
- Dynamic Actions
 - creating, [14-26](#)
 - debugging, [14-31](#)
 - deleting, [14-30](#)
 - editing, [14-29](#)
 - frequency and scope, [14-30](#)
 - managing, [14-22](#)
 - understanding, [14-23](#)
- dynamic list
 - about creating, [17-48](#)
 - creating from scratch, [17-50](#)
 - syntax, [17-48](#)
- dynamic translation, [21-61](#)

E

- Edit Application Definition
 - Allow Feedback, [6-30](#)
 - Application Alias, [6-28](#)
 - Application Groups, [6-29](#)
 - Build Status, [6-32](#)
 - Compatibility Mode, [6-30](#)
 - Debugging, [6-29](#)
 - Email from Address, [6-30](#)
 - Error Handling, Default Error Display
 - Location, [6-33](#)
 - Error Handling, Error Handling Function, [6-34](#)
 - Global Notifications, [6-34](#)
 - Logging, [6-29](#)
 - Message for unavailable application, [6-32](#)
 - Name attribute, [6-28](#)
 - Proxy Server, [6-31](#)
 - restrict access, [6-33](#)
 - Status, [6-32](#)
 - Version, [6-28](#)
- Edit Application Definition page
 - about, [6-27](#)
 - accessing, [6-27](#)
- Edit Application page
 - Database Session, [6-43](#)
- Edit Globalization Attributes, [6-44](#)
 - Application Date Format, [6-46](#)
 - Application Date Time Format, [6-46](#)

- Edit Globalization Attributes (*continued*)
 - Application Language Derived From, [6-46](#)
 - Application Primary Language, [6-45](#)
 - Application Timestamp Format, [6-46](#)
 - Application Timestamp Time Zone Format, [6-47](#)
 - Automatic CSV Encoding, [6-48](#)
 - Automatic Time Zone, [6-47](#)
 - Character Value Comparison, [6-47](#)
 - Character Value Comparison Behavior, [6-47](#)
 - Document Direction, [6-46](#)
- Edit Security Attributes, [6-35](#)
 - accessing, [6-36](#)
 - Authentication, [6-37](#)
 - Authorization, [6-38](#)
 - Session Management, [6-38](#)
 - Session State Protection, [6-41](#)
- editable interactive grid, [10-2](#)
- email
 - configuring for instance, [19-2](#)
 - email log, [19-1](#)
 - email queue, [19-1](#)
 - enabling, [20-13](#)
 - managing, [19-1](#)
 - sending with APEX_MAIL package, [19-3](#)
- Email from Address attribute, [6-30](#)
- Email Reporting
 - creating with Create Page Wizard, [7-7](#)
- email templates, [17-23](#)
 - creating, [17-24](#)
 - deleting, [17-28](#)
 - editing, [17-28](#)
 - including dynamic text, [17-27](#)
 - viewing samples, [17-26](#)
- Embed in Frames attribute, [6-42](#)
- embedded PL/SQL gateway
 - security practices, [20-5](#)
- engine, [1-3](#)
- error message
 - creating for a page process, [16-15](#)
 - creating for application processes, [16-20](#)
- errors, handling, [6-33](#)
- Expire Bookmarks attribute, [6-41](#)
- export
 - application, [24-27](#)
 - application attributes, [24-30](#)
 - build option status, [24-30](#)
 - database application, [24-25](#)
 - database application components, [24-25](#)
 - feedback, [24-37](#)
 - managing database objects, [24-6](#)
 - page, [24-32](#)
 - page components, [24-30](#)
 - plug-in, [24-34](#)
 - script file, [24-38](#)

- export (*continued*)
 - shared components, [24-30](#)
 - static files, [24-34](#)
 - text for translations, [21-9](#)
 - themes, [24-34](#)
 - User Interface Defaults, [24-35](#)
 - Websheet, [24-33](#)
 - workspaces, [24-31](#)
- Export Repository, [24-44–24-46](#)
 - accessing, [24-44](#)
- Export/Import icon, [1-24](#)
- exported application, importing, [24-38](#)

F

- f?p syntax, [2-14](#)
- feature reports
 - Approval Statuses, [4-12](#)
 - Calendar page, [4-10](#)
 - Focus Areas, [4-11](#)
 - History, [4-10](#)
 - Owners, [4-12](#)
 - Progress Log, [4-11](#)
 - Select a Utility, [4-12](#)
 - Tree page, [4-9](#)
- feature utilities
 - assigning milestones, [4-33](#)
 - change milestones for overdue features, [4-33](#)
 - push due date of open features, [4-33](#)
 - set feature due date to milestone, [4-33](#)
- features
 - attaching files, [4-7](#)
 - creating, [4-5](#)
 - deleting, [4-6](#)
 - removing, [6-11](#), [7-10](#), [20-51](#)
 - tracking, [4-4](#)
 - updating, [4-6](#)
 - viewing, [4-4](#)
 - viewing reports, [4-8](#)
- feedback
 - about Feedback, [4-24](#)
 - adding with Create Application Wizard, [4-25](#)
 - adding with Create Page Wizard, [4-26](#)
 - By Filing User, [4-29](#)
 - Calendar page, [4-29](#)
 - configuring feedback, [4-28](#)
 - configuring support for attachments, [4-28](#)
 - Feedback page, [4-27](#)
 - managing feedback, [4-28](#)
 - reviewing, [4-27](#)
 - submitting, [4-27](#)
 - track in Team Development, [4-28](#)
- Feedback
 - creating with Create Application Wizard, [6-8](#)
 - creating with Create Page Wizard, [7-7](#)

- field-level Help, [1-31](#)
 - viewing, [1-31](#)
 - viewing in Page Designer, [1-32](#)
- file upload table
 - security risks, [20-42](#)
- Find icon, [6-63](#)
- focus areas
 - renaming, [4-34](#)
- footer, substitution strings, [13-8](#)
- form
 - creating with Create App Wizard, [6-6](#)
 - creating with Create Page Wizard, [7-5](#)
- Form
 - legacy master detail form, [25-13](#)
- form on a table, [10-2](#)
- forms
 - creating a tabular form, [25-16](#)
 - creating Side by Side master detail, [10-17](#)
 - creating single page master detail, [10-18](#)
 - creating two page master detail, [10-20](#)
 - creating using Create Page Wizard, [10-4](#)
 - creating with Create Application Wizard, [10-8](#)
 - creating with Create Region Wizard, [10-10](#)
 - editable interactive grid, [10-2](#)
 - form on a table, [10-2](#)
 - managing master detail, [10-11](#)
 - master detail, [10-11](#)
 - understanding types, [10-1](#)
 - validating input, [10-21](#)
- full development environment, [1-4](#)
- functions, searching for, [6-66](#)

G

- Gallery
 - about, [8-13](#)
 - using, [8-25–8-27](#)
 - viewing legacy components, [25-21](#)
- Gallery menu
 - Show Legacy Components, [25-21](#)
- Global Notifications attribute, [6-34](#)
- Global page, [13-3](#)
 - creating, [13-4](#)
 - navigating to, [13-4](#)
- Global Search, [1-7](#)
- global template option group, [12-36](#)
 - creating, [12-36](#)
 - editing, [12-36](#)
- global template options
 - creating, [12-39](#)
 - editing, [12-39](#)
 - managing, [12-37](#)
- Global Templates, [12-16](#)
- global variable, creating, [17-40](#)

globalization
 attributes, [21-7](#)
 determining application language, [6-44](#)
 understanding, [21-2](#)

graphics
 managing, [13-15](#)

grid messages requiring translation
 interactive grid, [21-18](#)

groups
 of applications, [6-68](#)
 of database application pages, [7-16](#)

H

Help
 accessing, [1-29](#)
 creating, [11-41](#)
 creating navigation bar entry, [11-45](#)
 searching, [1-29](#)
 using, [1-29](#)

Help menu, [1-11](#)

Help text
 seeding, [11-45](#)

hidden item, [A-5](#)

HTML
 generated using custom PL/SQL, [13-15](#)
 not handled by Application Express, [13-15](#)

HTML Escaping Mode attribute, [6-42](#)

HTML password form element, [A-5](#)

HTML text area, [A-7](#)

HTML5 chart
 switching to Flash chart, [25-9](#)

HTTP Header Variable authentication
 setting up, [20-63](#)

HTTP Response Headers attribute, [6-43](#)

I

icon view
 interactive grids, [9-32](#)

Image Finder, [6-63](#)

Image Prefix attribute, [6-50](#)

Image Repository, [13-15](#)

image, displaying from URL, [A-3](#)

images
 managing, [13-15](#)
 searching for, [6-67](#)
 viewing available, [6-67](#)

Images Finder, [6-67](#)

implementing
 plug-ins, [19-4](#)

import
 feedback, [24-43](#)
 Websheet, [24-40](#)

importing
 application, [24-39](#)
 application using SQL*Plus, [24-46](#)
 export files, [24-38](#)
 page, [24-39](#)
 plug-in, [24-41](#)
 themes, [24-41](#)
 User Interface Defaults, [24-42](#)

Initialization PL/SQL Code, [6-43](#)

injection attacks, restricting characters on items,
[20-28](#)

installation
 enabling network services, [20-17](#)

installing
 application files, [24-44](#)
 export files, [24-45](#)

installing application export, [24-44](#)

installing export files, [24-45](#)

instance proxy, specifying, [20-7](#)

instance settings
 configuring, [20-13](#)
 configuring email, [19-2](#)

interactive grid
 creating with Create App Wizard, [6-6](#)
 creating with Create Application Wizard, [9-8](#)
 creating with Create Page Wizard, [10-6](#)
 managing, [9-18](#)
 message requiring translation, [21-18](#)

interactive grid messages, translating, [21-18](#)

interactive grids
 about, [9-2](#)
 controlling pagination, [9-28](#)
 creating a column link, [9-39](#)
 creating master detail form, [9-21](#)
 customizing in runtime, [9-41](#)
 customizing toolbar, [9-29](#)
 defining a column as list of values, [9-38](#)
 deleting a report, [9-45](#)
 displaying conditionally, [9-24](#)
 editing Attributes, [9-26](#)
 editing column attributes, [9-37](#)
 enabling Detail view, [9-35](#)
 enabling icon view, [9-32](#)
 enabling saving public reports, [9-30](#)
 making editable, [9-26](#)
 managing Attributes, [9-25](#)
 managing column attributes, [9-36](#)
 managing region attributes, [9-19](#)
 renaming saved report, [9-45](#)
 resetting a report, [9-45](#)
 restricting saving of Public reports, [9-31](#)
 saving, [9-41](#)
 saving and authentication, [9-42](#)
 saving configuration dependencies, [9-42](#)
 saving Primary report, [9-44](#)

- interactive grids (*continued*)
 - saving private report, [9-43](#)
 - saving public report, [9-43](#)
 - user save options, [9-42](#)
 - interactive report filters, [9-76](#)
 - editing, [9-76](#)
 - enabling and disabling, [9-76](#)
 - removing, [9-76](#)
 - showing and hiding details, [9-76](#)
 - interactive report messages, translating, [21-44](#)
 - interactive report regions
 - configuring print attributes, [9-106](#)
 - interactive reports
 - about, [9-4](#)
 - about link columns, [9-50](#)
 - about Search bar, [9-75](#)
 - Actions menu, [9-76](#)
 - configuring structure, [17-17](#)
 - adding download option, [9-58](#)
 - advanced attributes, [9-67](#)
 - Attributes, [9-49](#)
 - changing pagination, [9-63](#)
 - column attributes, [9-70](#)
 - configuring Actions menu structure, [17-17](#)
 - creating a column link, [9-71](#)
 - creating with Create Application Wizard, [9-7](#)
 - creating with REST Enabled SQL, [18-18](#)
 - customizing in runtime, [9-74](#)
 - defining a column as a list of values, [9-72](#)
 - deleting private report, [9-82](#)
 - deleting public report, [9-82](#)
 - displaying conditionally, [9-48](#)
 - Email From Address, [9-67](#)
 - enabling Icon view, [9-63](#)
 - exporting, [9-58](#)
 - filtering on conditional links, [9-74](#)
 - formatting columns, [9-73](#)
 - invalid linking examples, [9-86](#)
 - linking restrictions, [9-86](#)
 - linking to, [9-83](#)
 - linking with clear cache syntax, [9-85](#)
 - linking with name and value syntax, [9-84](#)
 - linking with Request syntax, [9-84](#)
 - managing Attributes, [9-49](#)
 - managing column attributes, [9-69](#)
 - managing region attributes, [9-47](#)
 - managing saved reports, [6-78](#)
 - managing subscriptions, [6-78](#)
 - message requiring translation, [21-44](#)
 - Region Alias, [9-67](#)
 - renaming private report, [9-82](#)
 - renaming public report, [9-82](#)
 - Report ID Item, [9-67](#)
 - saving, [9-77](#)
 - saving alternative, [9-81](#)
 - interactive reports (*continued*)
 - saving Alternative report, [9-44](#)
 - saving as private, [9-79](#)
 - saving as public, [9-79](#)
 - saving primary default, [9-81](#)
 - user save options, [9-78](#)
 - using filters, [9-76](#)
 - valid linking examples, [9-86](#)
 - internal messages, requiring translation, [21-39](#)
 - IP address, restricting access, [20-7](#)
 - isolating workspaces, about, [20-9](#)
 - item
 - naming conventions, [14-19](#)
 - referencing values, [14-20](#)
 - searching for, [6-64](#)
 - Item Finder, [6-63](#), [6-64](#)
 - item help
 - editing multiple topics, [11-44](#)
 - item Help
 - viewing, [1-31](#)
 - item type
 - check box, [A-2](#)
 - Color Picker, [A-2](#)
 - Date Picker, [A-3](#)
 - Display Image, [A-3](#)
 - Display Only, [A-3](#)
 - File Browse, [A-4](#)
 - Hidden, [A-5](#)
 - List Manager, [A-5](#)
 - Number Field, [A-5](#)
 - Password, [A-5](#)
 - Percent Graph, [A-5](#)
 - Popup LOV, [A-6](#)
 - Radio, [A-6](#)
 - Rich Text Editor, [A-7](#)
 - Select list, [A-7](#)
 - Shuttle, [A-8](#)
 - Slider, [A-8](#)
 - Switch, [A-8](#)
 - Text Area, [A-10](#)
 - Text Field, [A-9](#)
 - Text Field with autocomplete, [A-9](#)
- ## J
-
- JavaScript, [12-66](#)
 - adding using Page Designer, [8-32](#)
 - calling from button, [14-36](#)
 - in row templates, [12-80](#)
 - including in .js file, [12-33](#)
 - including in page template, [12-33](#)
 - incorporating, [14-44](#)
 - libraries, [12-66](#)
 - referencing items, [14-21](#)
 - using a dynamic action, [14-31](#)

JavaScript Finder, [6-63](#)
 JET Chart, [11-4](#)
 Job Reporting
 creating with Create Page Wizard, [7-7](#)
 jQuery
 support, [19-14](#)
 jQuery Mobile
 support, [6-23](#)
 jQuery UI
 support, [19-14](#)

L

label template
 #CURRENT_ITEM_HELP_TEXT#, [12-55](#)
 creating page item Help, [12-55](#)
 Definition, [12-54](#)
 editing, [12-53](#)
 Error Display, [12-54](#)
 Field Container, [12-55](#)
 Name, [12-53](#)
 Substitution Strings, [12-55](#)
 language
 defining for application, [21-7](#)
 multibyte, [21-8](#)
 preference, [21-3](#)
 layout
 controlling, [12-1](#)
 LDAP Directory Verification
 setting up, [20-64](#)
 legacy calendar, [25-20](#)
 legacy calendar template, [12-51](#)
 legacy chart
 creating refresh dynamic action, [25-10](#)
 legacy chart syntax, [25-3](#)
 legacy charts
 about Flash, [25-2](#)
 adding to a page, [25-7](#)
 adding to new page, [25-5](#)
 altering display, [25-8](#)
 candlestick syntax, [25-4](#)
 chart syntax, [25-3](#)
 dial chart syntax, [25-3](#)
 editing, [25-8](#)
 editing attributes, [25-8](#)
 enabling asynchronous updates, [25-10](#)
 Gantt syntax, [25-4](#)
 legacy HTML5 chart about, [25-2](#)
 multiple series syntax, [25-3](#)
 range syntax, [25-4](#)
 scatter syntax, [25-4](#)
 SQL query syntax, [25-2](#)
 legacy components
 viewing in Gallery, [25-21](#)
 legacy dial chart syntax, [25-3](#)

legacy Flash chart
 candlestick syntax, [25-4](#)
 Gantt syntax, [25-4](#)
 multiple series syntax, [25-3](#)
 range syntax, [25-4](#)
 saving locally, [25-11](#)
 scatter syntax, [25-4](#)
 using custom XML, [25-12](#)
 legacy HTML5 chart
 about, [25-2](#)
 multiple series syntax, [25-3](#)
 using custom XML, [25-12](#)
 Legacy pages
 creating with Create Page Wizard, [7-6](#), [7-9](#)
 limits
 forms, [B-1](#)
 interactive reports, [B-1](#)
 item names, [B-1](#)
 tabular forms, [B-1](#)
 validations, [B-1](#)
 Link Column
 about, [9-50](#)
 excluding, [9-54](#)
 linking a single row view, [9-51](#)
 linking to a page, [9-52](#)
 linking to URL, [9-53](#)
 linking
 deep, [20-57](#)
 links
 adding, [4-35](#)
 deleting, [4-35](#)
 editing, [4-35](#)
 list
 about creating, [17-40](#), [17-41](#)
 about dynamic, [17-48](#)
 about navigation menus, [17-41](#)
 about static, [17-42](#)
 adding to a page, [17-52](#)
 Conditional Entries report, [17-54](#)
 copying, [17-51](#)
 copying between applications, [17-51](#)
 copying from current application, [17-51](#)
 creating hierarchical, [12-56](#)
 editing, [17-53](#)
 editing attributes, [17-53](#)
 History report, [17-54](#)
 Lists page, [17-53](#)
 navigation menu
 about creating, [17-41](#)
 reports, [17-54](#)
 tab
 switching to navigation menu, [17-41](#)
 types, [17-41](#)
 Unused report, [17-54](#)
 Utilization report, [17-54](#)

- list of values
 - bulk edit of static LOVs, [17-13](#)
 - Conditional Entries report, [17-14](#)
 - creating, [17-11](#)
 - creating at application-level, [17-12](#)
 - Edit Null Text report, [17-14](#)
 - editing, [17-13](#)
 - Grid Edit report, [17-14](#)
 - History report, [17-14](#)
 - Locally Defined report, [17-14](#)
 - named, [17-12](#)
 - referencing session state, [17-14](#)
 - reports, [17-14](#)
 - shared, [17-12](#)
 - static, [17-12](#)
 - Subscription report, [17-14](#)
 - translating, [21-61](#)
 - Utilization report, [17-14](#)
- list reports
 - Conditional Entries, [17-54](#)
 - History, [17-54](#)
 - Unused, [17-54](#)
 - Utilization, [17-54](#)
- list template
 - After List Entry, [12-59](#)
 - After Sub List Entry, [12-59](#)
 - Before List Entry, [12-57](#)
 - Before Sub List Entry, [12-58](#)
 - editing, [12-56](#)
 - Name, [12-57](#)
 - Sub List Entry, [12-58](#)
 - Substitution Strings, [12-59](#), [12-64](#)
 - Template Definition, [12-57](#)
 - Template Definitions for First Entry, [12-58](#)
- lists of terms, [A-5](#)
- Lists page, [17-53](#)
- Lists Wizard, [17-42](#), [17-48](#)
- live template options
 - using, [12-40](#)
- locks
 - determining lock status, [7-19](#)
 - for database application page, [7-19](#)
- Logging attribute, [6-29](#)
- login controls, creating, [20-6](#)
- login page
 - creating, [20-71](#)
- Login Page
 - creating with Create Page Wizard, [7-7](#)
- logo
 - defining, [6-52](#)
- Logo attribute, [6-51](#)
- logs
 - managing, [22-4](#)
 - performance, [22-3](#)

M

- Manage Session State Protection attribute, [6-41](#)
- map chart
 - about support for, [11-21](#)
 - creating SQL queries, [11-21](#)
- maps, [11-21](#)
 - adding by running a wizard, [11-24](#)
 - adding in Page Designer, [11-25](#)
 - creating, [11-20](#)
 - editing map attributes, [11-26](#)
 - enabling automatic updates (Page Designer), [11-28](#)
 - managing in Page Designer, [11-26](#)
 - using custom XML, [11-27](#)
- master application, creating, [12-25](#)
- master detail
 - creating Side by Side, [10-17](#)
 - creating single page, [10-18](#)
 - creating two page, [10-20](#)
 - creating with Create App Wizard, [6-6](#)
 - creating with Create Page Wizard, [7-5](#), [10-17](#), [10-18](#), [10-20](#)
 - managing, [10-11](#)
- master detail form
 - creating with Create Application Wizard, [10-15](#)
- master detail form, about, [10-11](#)
- master detail forms
 - about, [10-3](#)
- master detail, single page
 - about, [10-3](#)
- master detail, two page
 - about, [10-3](#)
- Maximum Session Idle Time in Second attribute, [6-40](#)
- Maximum Session Length in Seconds attribute, [6-40](#)
- menu, [1-12](#)
- Message for unavailable application, [6-32](#)
- messages
 - creating for translation, [21-38](#)
 - editing text messages, [21-39](#)
 - in PL/SQL procedures, [21-17](#)
 - in reports
 - translating, [21-38](#)
 - translating, [21-16](#), [21-17](#)
 - translating internal, [21-38](#)
- messages in reports, [21-38](#)
- messages requiring translation
 - interactive reports, [21-44](#)
 - internal, [21-39](#)
- milestone reports
 - By Owner report, [4-15](#)
 - Feature by Milestone report, [4-15](#)

milestone reports (*continued*)
 Milestone Calendar, [4-14](#)

milestones
 creating, [4-13](#)
 deleting, [4-14](#)
 tracking, [4-13](#)
 updating, [4-14](#)
 viewing, [4-13](#)
 viewing reports, [4-14](#)

mobile applications
 about, [6-23](#)
 about designing, [6-22](#)
 creating, [6-22](#), [6-24](#)

monitor activity
 Calendar Reports, [22-6](#)
 Developer Activity, [22-6](#)
 Environment Reports, [22-6](#)
 Login Attempts, [22-6](#)
 Logs, [22-6](#)
 Page Views, [22-6](#)
 Service Requests, [22-6](#)
 Workspace Purge, [22-6](#)

multiple reports
 creating with Create App Wizard, [6-7](#)

multiple select list item, [14-21](#)
 handling return values, [14-21](#)
 using APEX_UTIL.STRING_TO_TABLE,
[14-22](#)

N

navigation
 branch, [15-23](#)
 breadcrumb, [15-8](#)
 list, [17-40](#)
 navigation menu, [17-55](#)
 parent tabs, [15-2](#)
 standard tabs, [15-2](#)
 tab, [15-2](#)

navigation bar
 creating from scratch, [15-18](#)

navigation bar list
 about, [17-57](#)
 editing, [17-58](#)
 managing, [17-57](#)

navigation menu
 about, [17-55](#)
 changing display, [17-56](#), [17-58](#)
 editing, [17-55](#)
 managing, [17-55](#)
 switching from tabs, [17-57](#)

network services
 enabling, [20-17](#)
 granting connect privileges, [20-18](#)

network services (*continued*)
 granting connect privileges, Oracle Database
 12c, [20-19](#)
 invalid ACL error, [20-20](#)

news entries, [1-6](#)
 adding, [4-35](#)
 deleting, [4-35](#)
 editing, [4-35](#)

number field, [A-5](#)

NV function, [2-18](#)

O

online Help, [1-29](#)
 searching, [1-29](#)

Open Door Credentials
 setting up, [20-67](#)

Oracle Application Express
 about, [1-2](#), [1-3](#)
 architecture, [1-3](#)
 component limits, [B-1](#)
 signing in, [1-2](#)
 user roles, [20-3](#)

Oracle Application Express administrator, [20-3](#)

Oracle Application Express Advisor
 about, [23-2](#)
 checking application integrity, [23-1](#)
 running on a page, [23-3](#)
 running on application, [23-2](#)

Oracle Application Express Engine, [1-3](#)

Oracle BI Publisher, integrating with, [20-16](#)

Oracle Help Center
 searching, [1-29](#), [1-30](#)

Oracle HTTP Server
 about security, [20-4](#)

Oracle JET
 about, [11-2](#)
 integrated with Oracle Application Express,
[11-2](#)
 integration, [11-1](#)

output escaping
 substitution strings, [2-25](#), [12-33](#)

P

packages, searching for, [6-66](#)

page attributes
 editing in Page Designer, [8-31](#)
 viewing in Page Designer, [8-32](#)
 Warn on Unsaved Changes, [8-36](#)

page components, controlling access, [20-50](#)

page computation
 computation point, [16-4](#)
 creating, [16-2](#)
 defining computation point, [16-3](#)

- page computation (*continued*)
 - defining computation source, [16-3](#)
 - editing attributes, [16-3](#)
 - page submission, [8-30](#)
 - pre-rendering, [8-30](#)
 - syntax, [16-4](#)
- Page Designer
 - accessing, [8-20](#)
 - accessing the Property Editor, [8-24](#)
 - adding a report, dragging and dropping, [9-17](#)
 - adding buttons, [8-25–8-27](#)
 - adding items, [8-25–8-27](#)
 - adding regions, [8-25–8-27](#)
 - Code Editor, [8-18](#)
 - Component View tab, [8-14](#)
 - configuring dialog attributes, [8-35](#)
 - copying buttons, [8-29](#)
 - copying items, [8-29](#)
 - copying page, [8-39](#)
 - copying regions, [8-29](#)
 - deleting application page, [8-40](#)
 - dragging and dropping tabs, [8-22](#)
 - editing page attributes, [8-32](#)
 - Gallery, [8-13](#)
 - Help, [8-16](#)
 - Layout, [8-11](#)
 - managing page attributes, [8-31](#)
 - marking region titles as translatable, [21-5](#)
 - Messages tab, [8-15](#)
 - Page Search tab, [8-16](#)
 - Property Editor, [8-17](#)
 - Reset Layout, [8-22](#), [8-23](#)
 - Right Pane, [8-17](#)
 - running a page, [8-37](#)
 - Runtime Developer toolbar, [8-38](#)
 - searching for pages, [8-21](#)
 - Three Pane mode, [8-23](#)
 - Tree pane, [8-7](#)
 - Two Pane mode, [8-23](#)
 - using, [8-20](#)
 - viewing field-level help, [1-32](#)
- Page Designer Toolbar, [8-4](#)
 - Create Form Region, [8-5](#)
 - Create menu, [8-4](#)
 - Create, Create Breadcrumb Region, [8-5](#)
 - Create, Developer Comment, [8-4](#)
 - Create, Page, [8-4](#)
 - Create, Page as Copy, [8-4](#)
 - Create, Page Group, [8-4](#)
 - Create, Shared Components, [8-4](#)
 - Create, Team Development, [8-4](#)
 - Developer Comment, Bug, or To Do button, [8-6](#)
 - Go To Page, [8-4](#)
 - Page Lock, [8-4](#)
- Page Designer Toolbar (*continued*)
 - Page Unlock, [8-4](#)
 - Redo, [8-4](#)
 - Save and Run Page button, [8-4](#)
 - Save button, [8-4](#)
 - Settings, Bugs, [8-6](#)
 - Settings, Features, [8-4](#)
 - Settings, Feedback Entries, [8-6](#)
 - Settings, To Dos, [8-4](#)
 - Settings, Two Pane Mode, [8-4](#)
 - Shared Components, [8-4](#)
 - Undo, [8-4](#)
 - Utilities, [8-4](#)
 - Utilities, Application Utilities, [8-4](#)
 - Utilities, Attribute Dictionary, [8-4](#)
 - Utilities, Delete, [8-4](#)
 - Utilities, Delete Page, [8-4](#)
 - Utilities, Export, [8-4](#)
 - Utilities, History, [8-4](#)
 - Utilities, Layout, [8-6](#)
 - Utilities, Page Groups, [8-4](#)
 - Utilities, Upgrade Application, [8-4](#)
- Page Designer, page attributes
 - Dialog: Attributes, [8-36](#)
 - Dialog: Chained, [8-36](#)
 - Dialog: CSS Classes, [8-36](#)
 - Dialog: Height, [8-36](#)
 - Dialog: Maximum Width, [8-36](#)
 - Dialog: Width, [8-36](#)
- Page Finder, [6-63](#)
- page item
 - types, [A-1](#)
- page layout, about, [13-1](#)
- page processes
 - changing processing points, [16-14](#)
 - changing source, [16-14](#)
 - creating, [16-12](#)
 - creating error message, [16-15](#)
 - creating success message, [16-15](#)
 - editing attributes, [16-14](#)
 - making conditional, [16-16](#)
 - page submission, [8-30](#)
 - pre-rendering, [8-30](#)
 - When Button Pressed attribute, [2-43](#)
- page processing, [2-6](#)
 - understanding, [2-6](#)
- page rendering, [2-6](#)
 - understanding, [2-6](#)
- Page Selector, [8-21](#)
- page specific utilities, [6-76](#)
 - Branch Utilities, [6-76](#)
 - Button Utilities, [6-76](#)
 - Computations Utilities, [6-76](#)
 - Cross Page Utilities, [6-76](#)
 - Dynamic Action Utilities, [6-76](#)

- page specific utilities (*continued*)
 - Item Utilities, [6-76](#)
 - Page Groups, [6-76](#)
 - Process Utilities, [6-76](#)
 - Region Utilities, [6-76](#)
 - Validation Utilities, [6-76](#)
- page template
 - Body, [12-60](#)
 - Cascading Style Sheet, [12-62](#)
 - Definition, [12-60](#)
 - Dialog, [12-63](#)
 - Display Points, [12-63](#)
 - editing, [12-59](#)
 - Error Page Template Control, [12-64](#)
 - Footer, [12-60](#)
 - Header, [12-60](#)
 - Image Based Tab Attributes, [12-62](#)
 - JavaScript, [12-61](#)
 - JavaScript, Dialog Cancel Code, [12-61](#)
 - JavaScript, Dialog Closure Code, [12-61](#)
 - JavaScript, Dialog Initialization Code, [12-61](#)
 - JavaScript, Execute when Page Loads, [12-61](#)
 - JavaScript, File URLs, [12-61](#)
 - JavaScript, Function and Global Variable Declaration, [12-61](#)
 - Layout, [12-63](#)
 - Name, [12-60](#)
 - selecting, [12-34](#)
 - specifying defaults, [12-34](#)
 - substitution strings, [12-59](#)
 - Subtemplate, [12-62](#)
- page template substitution strings, [12-59](#)
 - #APEX_CSS#, [12-66](#)
 - #APP_VERSION#, [12-64–12-66](#)
 - #APPLICATION_CSS#, [12-66](#)
 - #APPLICATION_JAVASCRIPT#, [12-66](#)
 - #BOX_BODY#, [12-64, 12-65](#)
 - #BUILT_WITH_LOVE_USING_APEX#, [12-66](#)
 - #DIALOG_CSS_CLASSES#, [12-67](#)
 - #DIALOG_HEIGHT#, [12-67](#)
 - #DIALOG_MAX_WIDTH#, [12-67](#)
 - #DIALOG_WIDTH#, [12-67](#)
 - #FAVICONS#, [12-66](#)
 - #FORM_CLOSE#, [12-65](#)
 - #FORM_OPEN#, [12-65](#)
 - #GLOBAL_NOTIFICATION#, [12-65](#)
 - #HEAD#, [12-66](#)
 - #IS_MODAL#, [12-67](#)
 - #LOGO#, [12-65](#)
 - #NAVIGATION_BAR#, [12-65](#)
 - #NOTIFICATION_MESSAGE#, [12-64, 12-65](#)
 - #ONLOAD#, [12-66](#)
 - #PAGE_TRANSITION#, [12-66](#)
- page template substitution strings (*continued*)
 - #REGION_POSITION_NN#, [12-64](#)
 - #SUCCESS_MESSAGE#, [12-65](#)
 - #TAB_CELLS#, [12-64, 12-65](#)
 - #TEMPLATE_JAVASCRIPT#, [12-65](#)
 - #THEME_STYLE_CSS#, [12-66](#)
 - #TITLE#, [12-64](#)
 - PAGE_URL, [12-66](#)
- page zero, [13-2](#)
- page-level item
 - about, [14-19](#)
 - applying format mask., [14-16](#)
 - configuring Escape Special Characters, [14-13](#)
 - configuring Restricted Characters, [14-14](#)
 - configuring security, [14-12](#)
 - configuring Session State Protection, [14-13](#)
 - configuring Store value encrypted in session state, [14-13](#)
 - creating, [14-2, 14-18](#)
 - creating a shuttle, [14-10](#)
 - creating cascading list of values (LOV), [14-8](#)
 - creating from Gallery, [14-3](#)
 - creating from Rendering tab, [14-3](#)
 - defining default values, [14-11](#)
 - defining quick picks, [14-14](#)
 - displaying as conditional, [14-15](#)
 - displaying as read only, [14-15](#)
 - editing in Property Editor, [14-11](#)
 - multiple select list item, [14-21](#)
 - naming conventions, [14-19](#)
 - referencing using JavaScript, [14-21](#)
 - referencing values, [14-20](#)
 - specifying Authorization Scheme, [14-13](#)
 - static list of values, [14-4](#)
 - viewing, [14-2](#)
 - viewing utilities, [14-18](#)
 - Warn on Unsaved Changes, [14-17](#)
- page-level item reports
 - All Page Items, [14-18](#)
 - Conditional Items, [14-18](#)
 - Grid Edit of all Item Help Text, [14-18](#)
 - Grid Edit of all Item Labels, [14-18](#)
 - Item Help Subscriptions, [14-18](#)
- page-level template, [12-34](#)
- page, attaching authorization scheme, [20-77](#)
- pages
 - about, [2-2](#)
 - about Global page, [13-3](#)
 - calling from a button, [2-16](#)
 - calling with an alias, [2-16](#)
 - controlling access to, [20-50](#)
 - enabling user customization, [13-8](#)
 - exporting, [24-32](#)
 - linking with f?p syntax, [2-14](#)

- pages (*continued*)
 - optimizing for printing, [13-2](#)
 - resource use, [23-3](#)
 - searching for, [6-65](#)
 - specifying page template, [12-34](#)
- Pages Finder, [6-65](#)
- pagination
 - classic report, [9-92](#)
 - interactive reports, [9-63](#)
 - reports, [12-74](#), [12-79](#)
- Pagination Subtemplate
 - Next Page Template, [12-80](#)
 - Next Set Template, [12-80](#)
 - Pagination Template, [12-80](#)
 - Previous Page Template, [12-80](#)
 - Previous Set Template, [12-80](#)
- parent tabs, creating, [15-2](#)
- Parsing Schema, [6-43](#)
- password
 - best practices, [20-22](#)
 - identifying risks, [20-23](#)
 - restricting reuse, [20-11](#)
 - Security Profiles Report, [20-23](#)
- passwords
 - Password Items Report, [20-23](#)
- patch, creating, [24-30](#)
- Percent Graph, [A-5](#)
- performance, [22-1](#)
 - best practices, [22-1](#)
 - database parameters, [22-6](#), [22-7](#)
 - debugging problematic SQL queries, [22-3](#)
 - diagnosing issues, [22-3](#)
 - identifying issues, [22-2](#)
 - limiting resources, [22-7](#)
 - slow queries, [22-3](#)
 - slow running applications, [22-2](#)
 - uploading static files to Web server, [22-8](#)
 - using logs and report, [22-3](#)
- PL/SQL code, searching for, [6-66](#)
- PL/SQL Finder, [6-63](#)
- plug-ins
 - about, [19-4](#)
 - accessing, [19-5](#)
 - adding events, [19-10](#)
 - creating, [19-5](#)
 - creating with Create Page Wizard, [7-5](#)
 - custom attributes, [19-7](#)
 - deleting, [19-10](#)
 - editing, [19-7](#)
 - examples, [19-4](#)
 - exporting, [19-12](#)
 - implementing, [19-4](#)
 - importing, [19-11](#), [24-41](#)
 - resetting, [19-13](#)
 - uploading associated files, [19-8](#)
- plug-ins (*continued*)
 - utilization, [19-13](#)
 - viewing history, [19-13](#)
 - viewing Plug-in Repository, [19-11](#)
- Popup LOV, [A-6](#)
- Popup LOV template
 - Buttons, [12-69](#)
 - editing, [12-68](#)
 - Page Attributes, [12-69](#)
 - Pagination, [12-69](#)
 - Result Set, [12-69](#)
 - Search Field, [12-68](#)
 - Window, [12-69](#)
- preconfigured authentication schemes, [20-59](#)
- Printer Friendly template, [13-2](#)
- printing reports, [9-102](#)
- privileges
 - about granting to database users, [D-1](#)
 - granted to database users, [D-1](#)
- Procedure Finder, [6-66](#)
- procedures, searching for, [6-66](#)
- process
 - Cache Management, [16-12](#)
 - Data Manipulation, [16-12](#)
 - Form Pagination, [16-12](#)
 - On Demand, [16-12](#)
 - PL/SQL, [16-12](#)
 - Report Pagination, [16-12](#)
 - Web Services, [16-12](#)
- productivity and aample App
 - installing, [3-5](#)
- productivity and sample app
 - installing, [3-4](#)
 - updating, [3-7](#)
- productivity and sample App
 - changing authentication, [3-6](#)
 - removing, [3-5](#)
- Productivity and Sample Apps Only workspace, [3-42](#)
 - accessing, [3-43](#)
 - account expiration, [3-39](#)
 - App Administration, [3-37](#), [3-43](#)
 - App Gallery, [3-43](#)
 - Dashboard, [3-43](#)
 - definition, [3-42](#)
 - end user account lifetime, [3-39](#)
 - locking, [3-39](#)
 - managing users, [3-39](#)
 - maximum login failures, [3-39](#)
 - requesting, [3-42](#)
 - viewing activity reports, [3-41](#)
 - viewing page views, [3-41](#)
 - viewing top users, [3-41](#)
- productivity application, unlocking, [3-6](#)

Productivity or Sample Applications
 about, [3-3](#)
 programming techniques
 collections, [18-8](#)
 implementing Web services, [18-40](#)
 proxy server, [18-42](#)
 Proxy Server attribute, [6-31](#)
 public file upload, enabling, [20-7](#)
 public synonyms, [D-1](#)
 Public User attribute, [6-37](#)

Q

Query Finder, [6-63](#), [6-65](#)
 query, searching for, [6-65](#)

R

radio group, [A-6](#)
 Recently Updated Pages report, [6-74](#)
 region
 adding an image, [13-11](#)
 adding sliders to regions, [17-17](#)
 based on a URL, [13-14](#)
 controlling positioning, [13-8](#)
 copying, [13-13](#)
 creating, [13-5](#)
 creating for a list, [17-52](#)
 creating region display selector, [13-12](#)
 deleting, [13-13](#)
 editing attributes, [13-7](#)
 editing region attributes, [13-7](#)
 enabling customization, [13-8](#)
 enabling users to customize a page, [13-8](#)
 specifying a static region ID, [13-10](#)
 specifying header and footer, [13-8](#)
 understanding, [13-4](#)
 region attributes
 Cache, [13-9](#)
 Static ID, [13-10](#)
 region display selector, creating, [13-12](#)
 region template
 Definition, [12-70](#)
 editing, [12-69](#)
 Form Table Attributes, [12-71](#)
 Name, [12-70](#)
 Sub Regions Attributes, [12-71](#)
 Substitution Strings, [12-72](#)
 region titles
 marking as translatable, [21-5](#)
 Rejoin Sessions
 about security risks, [20-9](#)
 configuring at application-level, [6-39](#)
 configuring at page-level, [8-34](#)
 Rejoin Sessions attribute, [6-39](#)

remote database
 accessing from existing components, [18-25](#)
 Remote Server
 about, [18-33](#)
 creating, [18-34](#)
 deleting, [18-35](#)
 editing, [18-35](#)
 exporting, [18-33](#)
 viewing history, [18-36](#)
 viewing utilization, [18-36](#)
 report
 based on spreadsheet, [6-13](#), [6-14](#)
 creating with Create App Wizard, [6-6](#)
 creating with Create Page Wizard, [7-5](#)
 Oracle database, [22-6](#)
 report column templates
 colorizing cell headings, [12-74](#)
 omitting headings, [12-74](#)
 report column templates (generic)
 After Rows, [12-76](#)
 Before Each Row, [12-74](#), [12-75](#)
 Before Rows, [12-74](#)
 Column Headings, [12-74](#)
 Column Templates, [12-75](#)
 displaying HTML after all columns, [12-75](#)
 displaying HTML after last row, [12-76](#)
 Next Page Template, [12-77](#)
 Next Set Template, [12-77](#)
 Pagination Subtemplate, [12-76](#)
 Previous Page Template, [12-77](#)
 Previous Set Template, [12-77](#)
 Report Template, [12-73](#)
 Row Highlighting, [12-76](#)
 report column templates (named)
 Before Rows attribute, [12-79](#)
 colorizing column header cell, [12-79](#)
 Column Headings, [12-79](#)
 Pagination Subtemplate, [12-79](#)
 Report Template Identification, [12-77](#)
 Row Templates, [12-78](#)
 Subscription, [12-78](#)
 report layouts
 about, [17-63](#)
 copying, [17-65](#)
 creating, [17-64](#)
 editing, [17-65](#)
 options, [17-64](#)
 report on a form on table, [9-6](#)
 report queries
 about, [17-59](#)
 copying, [17-62](#)
 creating, [17-60](#)
 editing, [17-62](#)
 report regions
 about configuration options, [9-103](#)

- report regions (*continued*)
 - printing, [9-102](#)
 - printing to PDF, [9-103](#)
 - report templates, [12-72](#)
 - editing, [12-72](#)
 - generic column templates, [12-72](#)
 - named column templates, [12-73](#)
 - using JavaScript, [12-80](#)
 - reports, [12-42](#)
 - about, [9-1](#)
 - Active Session History (ASH) reports, [22-6](#)
 - adding by dragging and dropping, [9-17](#)
 - Application Dashboard, [6-73](#), [22-5](#)
 - Automatic Workload Repository (AWR) reports, [22-6](#)
 - Change History, [6-75](#)
 - class references, [12-30](#)
 - creating with Create Application Wizard, [9-6](#), [9-8](#)
 - creating with Create Page Wizard, [9-11](#), [9-12](#)
 - CSV export, [12-74](#)
 - exporting to CSV format, [12-74](#), [12-79](#)
 - including pagination above, [12-74](#), [12-79](#)
 - managing interactive grids, [9-18](#)
 - monitor activity, [22-6](#)
 - performance, [22-3](#)
 - printing, [9-102](#)
 - report and form, [9-6](#)
 - supported substitution strings, [12-31](#)
 - template subscription, [12-42](#)
 - template utilization, [12-42](#)
 - templates in a theme, [12-29](#)
 - theme file references, [12-30](#)
 - theme template counts, [12-29](#)
 - themes, [12-29](#)
 - understanding types, [9-2](#)
 - workspace activity, [22-5](#)
 - REQUEST
 - about scope and value, [2-43](#)
 - about When Button Pressed attribute, [2-43](#)
 - button names, [14-37](#)
 - referencing using declarative conditions, [2-43](#)
 - referencing value of, [2-42](#)
 - substitution string, [2-42](#)
 - using for Show processing, [2-44](#)
 - resource use, monitoring, [23-3](#)
 - REST Enabled SQL
 - creating calendars, [18-23](#)
 - creating charts, [18-20](#)
 - creating interactive reports, [18-18](#)
 - editing existing components, [18-25](#)
 - REST Enabled SQL Reference, [18-9](#)
 - REST Enabled SQL service
 - creating reference, [18-12](#), [18-13](#)
 - deleting, [18-17](#)
 - REST Enabled SQL service (*continued*)
 - editing, [18-15](#)
 - requirements, [18-10](#)
 - REST Enabled SQL Services
 - about, [18-9](#)
 - exporting, [18-9](#)
 - RESTful access, enabling, [20-8](#)
 - restrict access, to application, [6-33](#)
 - Rich Text Editor
 - responsiveness, [17-18](#)
 - Run Application button, [1-24](#)
 - Run on Public Pages attribute, [6-38](#)
 - Runtime API Usage, [6-44](#)
 - Runtime Developer toolbar, [8-38](#)
 - Application, [8-38](#)
 - Debug, [8-38](#)
 - Developer Toolbar Options, [8-38](#)
 - Edit Page, [8-38](#)
 - Home, [8-38](#)
 - Page Info, [8-38](#)
 - Quick Edit, [8-38](#), [12-40](#)
 - Session, [8-38](#)
 - View Debug, [8-38](#)
 - runtime environment, [1-3](#)
 - advantages, [20-17](#)
- ## S
-
- safe item types, [20-25](#)
 - sample charts
 - viewing sample, [11-30](#)
 - Sample Database Application
 - about, [3-8](#)
 - Administration page, [3-18](#)
 - Customers page, [3-12](#)
 - editing, [3-19](#), [3-20](#)
 - editing an application, [3-20](#)
 - Home page, [3-10](#)
 - installing, [3-8](#)
 - modifying, [3-19](#), [3-20](#)
 - Navigation Menu, [3-11](#)
 - running, [3-8](#)
 - understanding, [3-8](#), [3-9](#)
 - Save Public Report Authorization, [9-31](#)
 - SDLC
 - rapid application development, [24-4](#)
 - Spiral, [24-4](#)
 - secure sockets layer, best practices, [20-8](#)
 - security
 - attributes, [6-35](#)
 - security attributes
 - Allows URLs Created After, [6-41](#)
 - authentication, [6-37](#)
 - Authentication Scheme, [6-37](#)
 - Authorization Scheme, [6-38](#)

- security attributes (*continued*)
 - Bookmark Hash Function, [6-41](#)
 - Browser Security, [6-42](#)
 - Cache, [6-42](#)
 - Database Session, [6-43](#), [6-44](#)
 - Deep Linking, [6-39](#)
 - Define Authorization Schemes, [6-38](#)
 - Embed in Frames, [6-42](#)
 - Expire Bookmarks, [6-41](#)
 - HTML Escaping Mode, [6-42](#)
 - HTTP Response Headers, [6-43](#)
 - Manage Session State Protection, [6-41](#)
 - Maximum Session Idle Time in Seconds, [6-40](#)
 - Maximum Session Length in Seconds, [6-40](#)
 - Public User, [6-37](#)
 - Rejoin Sessions, [6-39](#)
 - Run on Public Pages, [6-38](#)
 - Session Idle Timeout URL, [6-40](#)
 - Session Management, [6-38](#)
 - Session State Protection, [6-41](#)
 - Session Timeout URL, [6-40](#)
- security best practices
 - configuring instance settings, [20-13](#)
 - configuring storage options, [20-14](#)
 - configuring workspace settings, [20-14](#)
 - controlling SQL access in Websheets, [20-12](#)
 - creating login controls, [20-6](#)
 - creating wallets, [20-14](#)
 - cross site-scripting protection, [20-24](#)
 - embedded PL/SQL gateway, [20-5](#)
 - enabling email, [20-13](#)
 - enabling public file upload, [20-7](#)
 - enabling RESTful access, [20-8](#)
 - enabling RESTful services, [20-12](#)
 - for administrators, [20-1](#)
 - for developers, [20-21](#)
 - identifying risk passwords, [20-23](#)
 - isolating workspaces, [20-9](#)
 - items of type password, [20-22](#)
 - managing component availability, [20-16](#)
 - managing instance security, [20-5](#)
 - managing users, [20-15](#)
 - managing workspace to schema assignments, [20-15](#)
 - managing workspaces, [20-15](#)
 - Oracle BI Publisher, [20-16](#)
 - Oracle HTTP Server, [20-4](#)
 - restricting access by IP address, [20-7](#)
 - restricting password reuse, [20-11](#)
 - runtime environment, [20-17](#)
 - secure sockets layer, [20-8](#)
 - session joins, [20-9](#)
 - session state, [20-29](#)
 - session timeout, [20-10](#)
- security best practices (*continued*)
 - specifying instance proxy, [20-7](#)
- security best practicessecurity best practices
 - automating workspace provisioning, [20-13](#)
 - self service provisioning, [20-13](#)
- security risks, file upload table, [20-42](#)
- seeding, [21-9](#)
- select list, [A-7](#)
- session cloning
 - enabling, [2-22](#)
- session ID, [2-10](#)
- Session Idle Timeout URL, [6-40](#)
- session joins
 - preventing browser attacks, [20-9](#)
 - security best practices, [20-9](#)
- Session Management, [6-38](#)
 - specifying, [6-38](#)
- session state
 - clearing, [2-19](#)
 - clearing application cache, [2-20](#)
 - clearing cache by item, [2-19](#)
 - clearing cache by page, [2-19](#)
 - clearing cache for current user session, [2-21](#)
 - enabling projection, [6-41](#)
 - management, [2-10](#)
 - referencing, [2-17](#)
 - resetting an application completely, [2-20](#)
 - security best practices, [20-29](#)
 - sharing across applications, [17-40](#)
- session state protection
 - configuring all attributes, [20-33](#)
 - configuring security attributes, [20-32](#)
 - disabling, [20-30](#)
 - enabling, [20-30](#)
 - enabling from Edit Security Attributes, [20-30](#)
 - enabling from Session State Protection, [20-31](#)
 - enabling using a wizard, [20-31](#)
 - preventing URL tampering, [20-30](#)
 - reports, [20-32](#)
 - understanding, [20-30](#)
 - viewing existing settings, [20-32](#)
 - viewing summaries, [20-32](#)
- Session State Protection attribute, [6-41](#)
- session state values, managing, [2-17](#)
- Session Timeout URL, [6-40](#)
- session timeout, utilizing, [20-10](#)
- shared components
 - about, [17-2](#)
 - accessing, [17-2](#)
 - backing up, [24-30](#)
 - creating, [17-2](#)
 - exporting, [24-30](#)
- Shared Components icon, [1-24](#)

- Shared Components page
 - Application Access Control, [17-4](#)
 - Application Computations, [17-3](#)
 - Application Definition Attributes, [17-3](#)
 - Application Items, [17-3](#)
 - Application Processes, [16-16](#), [17-3](#)
 - Application Settings, [17-3](#)
 - Authentication Schemes, [17-4](#)
 - Authorization Schemes, [17-4](#)
 - Breadcrumbs, [17-6](#)
 - Component Settings, [17-5](#)
 - Credentials, [17-4](#), [17-9](#)
 - Data Load Definitions, [17-7](#)
 - Edit Application Comments, [17-8](#)
 - Email Templates, [17-5](#)
 - Export Application Components, [17-8](#)
 - Globalization Attributes, [17-8](#)
 - Legacy Web Service References (SOAP), [17-7](#)
 - List of Values, [17-4](#)
 - Lists, [17-5](#)
 - Manage Supporting Objects, [17-8](#)
 - Navigation Bar List, [17-6](#)
 - Navigation Menu, [17-5](#)
 - Plug-ins, [17-5](#)
 - Remote Servers, [17-9](#)
 - Report Layouts, [17-7](#)
 - Report Queries, [17-7](#)
 - REST Enabled SQL, [17-7](#), [17-9](#)
 - Review Developer Comments, [17-8](#)
 - Security Attributes, [17-4](#)
 - Session State Protection, [17-4](#)
 - Shortcuts, [17-5](#)
 - Static Application Files, [17-6](#)
 - Static Workspace Files, [17-6](#), [17-9](#)
 - Tabs, [17-5](#)
 - Templates, [17-6](#)
 - Text Messages, [17-8](#)
 - Themes, [17-6](#)
 - Translate Application, [17-8](#)
 - User Interface Attributes, [17-6](#)
 - Web Source Modules, [17-7](#)
- shared interactive reports, linking to, [9-87](#)
- shortcuts, [17-19](#)
 - about, [17-20](#)
 - copying an existing shortcut, [17-22](#)
 - defining, [17-21](#), [17-22](#)
 - defining from scratch, [17-21](#)
 - editing attributes, [17-22](#)
 - History report, [17-23](#)
 - HTML Text, [17-20](#), [17-21](#)
 - HTML Text with Escaped Special Characters, [17-20](#), [17-21](#)
 - Image, [17-20](#), [17-21](#)
 - Message, [17-20](#), [17-22](#)
 - shortcuts (*continued*)
 - Message with JavaScript Escaped Special Quotes, [17-20](#), [17-22](#)
 - PL/SQL Function Body, [17-20](#), [17-21](#)
 - referencing, [17-20](#)
 - reports, [17-23](#)
 - Subscription report, [17-23](#)
 - supporting translatable messages, [21-4](#)
 - syntax, [17-20](#)
 - Text with JavaScript Escaped Single Quotes, [17-20](#), [17-21](#)
 - Show Page, [2-6](#)
 - shuttle multiple select list, [A-8](#)
 - sign in page, [1-2](#)
 - Single Sign-On
 - setting up, [20-68](#)
 - Slider item type, [A-8](#)
 - SOAP, [18-40](#)
 - Social Sign-In Authentication, [20-69](#)
 - spreadsheet
 - creating app from, [6-13](#), [6-14](#)
 - SQL tracing, enabling, [23-4](#)
 - stacked master detail
 - creating with Create Page Wizard, [10-7](#)
 - standard tabs, creating, [15-2](#)
 - static application files
 - deleting, [17-32](#)
 - deleting all, [17-33](#)
 - downloading, [17-31](#)
 - downloading in Zip file, [17-32](#)
 - managing, [17-29](#)
 - referencing using #APP_IMAGES#, [17-31](#)
 - uploading, [17-30](#)
 - viewing details, [17-30](#)
 - static files
 - exporting, [24-34](#)
 - importing into another instance, [24-34](#)
 - static list
 - about, [17-42](#)
 - adding list entries, [17-43](#)
 - adding sublists, [17-43](#)
 - copying entries, [17-46](#)
 - creating from scratch, [17-43](#)
 - managing orphaned list entries, [17-47](#)
 - reparenting entries, [17-46](#)
 - resequencing entries, [17-47](#)
 - static substitution string, [6-35](#)
 - static workspace files
 - deleting, [17-36](#)
 - deleting all, [17-36](#)
 - downloading, [17-35](#)
 - downloading all in Zip file, [17-35](#)
 - managing, [17-33](#)
 - referencing using #WORKSPACE_IMAGES#, [17-34](#)

- static workspace files (*continued*)
 - uploading, [17-33](#)
 - viewing details, [17-34](#)
 - Status attribute, [6-32](#)
 - storage options, configuring, [20-14](#)
 - style sheet, [12-32](#)
 - substitution strings, [2-30](#)
 - #CSV_LINK#, [12-74](#), [12-79](#)
 - #GLOBAL_NOTIFICATION#, [6-34](#)
 - about built-in, [2-26](#)
 - APEX\$ROW_NUM, [2-28](#)
 - APEX\$ROW_SELECTOR, [2-28](#)
 - APEX\$ROW_STATUS, [2-28](#)
 - APP_ALIAS, [2-29](#)
 - APP_BUILDER_SESSION, [2-30](#)
 - APP_DATE_TIME_FORMAT, [2-31](#)
 - APP_ID, [2-29](#)
 - APP_IMAGES, [2-31](#)
 - APP-NLS_DATE_FORMAT, [2-31](#)
 - APP-NLS_TIMESTAMP_FORMAT, [2-32](#)
 - APP-NLS_TIMESTAMP_TZ_FORMAT, [2-32](#)
 - APP_PAGE_ALIAS, [2-33](#)
 - APP_PAGE_ID, [2-33](#)
 - APP_REQUEST_DATA_HASH, [2-33](#)
 - APP_SESSION, [2-34](#)
 - APP_SESSION_VISIBLE, [2-34](#)
 - APP_TEXT\$Message_Name,
 - APP_TEXT\$Message_Name\$Lang, [2-41](#)
 - APP_TITLE, [2-35](#)
 - APP_UNIQUE_PAGE_ID, [2-35](#)
 - APP_USER, [2-36](#)
 - Application Express SCHEMA OWNER, [2-44](#)
 - AUTHENTICATED_URL_PREFIX, [2-37](#)
 - BROWSER_LANGUAGE, [2-37](#)
 - calendar, [25-21](#)
 - CURRENT_PARENT_TAB_TEXT, [2-37](#)
 - DEBUG, [2-38](#)
 - for application items, [2-23](#)
 - for page items, [2-23](#)
 - IMAGE_PREFIX, [2-39](#)
 - in page templates, [12-59](#)
 - JET_BASE_DIRECTORY, [2-39](#)
 - JET_CSS_DIRECTORY, [2-39](#)
 - JET_JS_DIRECTORY, [2-40](#)
 - LOGOUT_URL, [2-40](#)
 - output escaping, [2-25](#), [12-33](#)
 - PRINTER_FRIENDLY, [2-41](#)
 - PROXY_SERVER, [2-41](#)
 - PUBLIC_URL_PREFIX, [2-42](#)
 - report of supported, [12-31](#)
 - REQUEST, [2-42](#)
 - SQLERRM, [2-44](#)
 - static, [6-35](#)
 - supported in region footer, [13-8](#)
 - substitution strings (*continued*)
 - SYSDATE_YYYYMMDD, [2-45](#)
 - THEME_DB_IMAGES, [2-45](#)
 - THEME_IMAGES, [2-45](#)
 - understanding, [2-23](#)
 - usage within templates, [2-26](#)
 - using, [2-23](#)
 - within templates, [2-23](#)
 - WORKSPACE_ID, [2-46](#)
 - WORKSPACE_IMAGES, [2-45](#)
 - success message, creating for a page process, [16-15](#)
 - Supported Charts
 - Oracle JavaScript Extension Toolkit (Oracle JET), [11-29](#)
 - supporting objects
 - creating, [24-9](#)
 - deinstalling, [24-15](#)
 - Supporting Objects icon, [1-24](#)
 - Switch
 - configuring, [17-18](#)
 - Switch item type, [A-8](#)
 - syntax, for dynamic list, [17-48](#)
 - system message, [1-6](#)
- ## T
-
- tab
 - about Tabs page, [15-3](#)
 - adding, [15-2](#)
 - Conditional Display report, [15-8](#)
 - create Standard tab, [15-7](#)
 - create Standard tab set, [15-7](#)
 - creating, [15-3](#), [15-5](#)
 - editing, [15-7](#)
 - editing multiple, [15-7](#)
 - History report, [15-8](#)
 - managing, [15-3](#)
 - move standard to different parent, [15-7](#)
 - Parent, [15-2](#)
 - rename standard, [15-7](#)
 - reports, [15-8](#)
 - resequence, [15-7](#)
 - Standard, [15-2](#)
 - Standard Tab Task, [15-7](#)
 - template support, [15-4](#)
 - Utilization report, [15-8](#)
 - tab reports
 - Conditional Display, [15-8](#)
 - History, [15-8](#)
 - Table Finder, [6-63](#)
 - Tables Finder, [6-65](#)
 - tables, searching for, [6-65](#)
 - Tabs page, [15-3](#)

- tabular form
 - creating, [25-16](#)
- Team Development
 - accessing, [4-2](#)
 - Feature Approval Statuses, [4-12](#)
 - Feature Focus Areas, [4-11](#)
 - Feature History, [4-10](#)
 - Feature Progress Log, [4-11](#)
 - Features Calendar, [4-10](#)
 - Features dashboard, [4-12](#)
 - Features Tree, [4-9](#)
 - home page, [4-2](#)
 - managing bugs, [4-19](#)
 - managing feedback, [4-24](#)
 - managing news entries, [4-35](#)
 - managing To Dos, [4-15](#)
 - News Region, [4-2](#)
 - purging data, [4-34](#)
 - To Dos Calendar, [4-18](#)
 - tracking features, [4-4](#)
 - tracking milestones, [4-13](#)
 - Utilities, [4-2](#)
- Team Development file attachments
 - downloading, [4-34](#)
- Team Development utilities, [4-30](#)
 - accessing, [4-31](#)
 - assigning milestones, [4-33](#)
 - configuring Team Development settings, [4-32](#)
 - downloading file attachments, [4-34](#)
 - enable support for file upload, [4-31](#)
 - manage focus areas, [4-31](#)
 - manage links, [4-31](#)
 - manage news, [4-31](#)
 - manage uploaded files, [4-31](#)
 - managing news entries, [4-35](#)
 - Purge Data, [4-31](#)
 - purging data, [4-34](#)
 - push due date of open features, [4-33](#)
 - push past due bugs, [4-31](#)
 - Release Summary, [4-33](#)
 - Release Summary report, [4-31](#)
 - renaming focus areas, [4-34](#)
 - set feature due date to milestone, [4-33](#)
 - Team Development settings, [4-31](#)
 - update assignee, [4-31](#)
- template edit history, [12-42](#)
- template option presets, [12-10](#)
- template options
 - about, [12-9](#)
 - default, [12-10](#)
 - groups, [12-10](#)
 - presets, [12-10](#)
 - using, [12-9](#)
- template options groups, [12-10](#)
- templates
 - breadcrumbs, [12-46](#)
 - button, [12-49](#)
 - changing defaults, [12-11](#)
 - columns, [12-75](#), [12-80](#)
 - creating, [12-35](#)
 - customizing, [12-32](#)
 - labels, [12-53](#)
 - legacy calendar, [12-51](#)
 - lists, [12-56](#)
 - managing, [12-43](#)
 - page, [12-59](#)
 - popup LOV, [12-68](#)
 - publishing, [12-46](#)
 - regions, [12-69](#)
 - replacing, [12-44](#)
 - replacing all, [12-45](#)
 - reports, [12-72](#)
 - selecting for a page, [12-34](#)
 - selecting theme default, [12-34](#)
 - supported substitution strings, [12-31](#)
 - unsubscribing to, [12-46](#)
 - viewing for specific page, [12-44](#)
 - viewing History report, [12-42](#)
 - viewing on Templates page, [12-43](#)
 - viewing Publish report, [12-43](#)
 - viewing region position utilization, [12-45](#)
 - viewing reports, [12-42](#)
 - viewing Subscription report, [12-42](#)
 - viewing Utilization report, [12-42](#)
- text area item type, [A-10](#)
- Text Field item type, [A-9](#)
- text strings, [21-38](#)
 - translating, [21-17](#)
- text strings, translating, [21-16](#)
- Text Text Field with autocomplete item type, [A-9](#)
- theme attributes, [12-11](#)
 - Breadcrumb, [12-14](#), [12-15](#)
 - Button, [12-14](#)
 - Charts, [12-15](#)
 - Classic Report, [12-14](#)
 - CSS File URLs, [12-13](#)
 - Custom Classes, [12-16](#)
 - Custom Image, [12-16](#)
 - Custom Library File URLs, [12-16](#)
 - Custom Prefix Class, [12-16](#)
 - Date Picker Icon Attributes, [12-16](#)
 - Date Picker Icon Name, [12-16](#)
 - Default Label, [12-14](#)
 - Dialog Button Region, [12-15](#)
 - Dialog Content Region, [12-15](#)
 - Dialog Page, [12-15](#)
 - Error Page, [12-14](#)
 - Files, [12-17](#)
 - Footer Toolbar, [12-14](#)

- theme attributes (*continued*)
 - Forms, [12-15](#)
 - Global Template Options, [12-16](#)
 - Header Toolbar, [12-14](#)
 - Interactive Reports, [12-15](#)
 - JavaScript File URLs, [12-13](#)
 - Legacy Calendar, [12-14](#)
 - Library, [12-16](#)
 - List, [12-14](#)
 - Lists, [12-15](#)
 - Login Page, [12-13](#)
 - Name, Application, [12-12](#)
 - Name, Description, [12-12](#)
 - Name, Name, [12-12](#)
 - Name, Navigation Bar Implementation, [12-12](#)
 - Name, Navigation Type, [12-12](#)
 - Name, Theme Number, [12-12](#)
 - Name, User Interface, [12-12](#)
 - Navigation Bar List, [12-13](#)
 - Navigation Menu List (Top), [12-13](#)
 - Navigation Menu List Position, [12-13](#)
 - Optional Label, [12-14](#)
 - Page, [12-13](#)
 - Page Template Options, [12-14](#)
 - Page Transition, [12-15](#)
 - Popup Transition, [12-15](#)
 - Printer Friendly Page, [12-14](#)
 - Region, [12-14](#)
 - Report, [12-15](#)
 - Required Label, [12-14](#)
 - Styles, [12-17](#)
 - Tabular Forms, [12-15](#)
 - Theme Subscription, [12-12](#)
 - Wizards, [12-15](#)
- theme reports
 - Application Templates, [12-29](#)
 - class references, [12-30](#)
 - Class References, [12-30](#)
 - File References, [12-30](#)
 - files references, [12-30](#)
 - supported substitution strings, [12-31](#)
 - template counts, [12-29](#)
 - Template Substitution Strings, [12-31](#)
 - templates in a theme, [12-29](#)
 - Theme Template Counts, [12-29](#)
- Theme Roller, [8-38](#)
 - using, [12-6](#)
- Theme Style Selection
 - creating with Create Application Wizard, [6-8](#)
 - creating with Create Page Wizard, [7-8](#)
- theme styles, [12-4](#)
 - about, [12-4](#)
 - creating, [12-4](#)
 - deleting, [12-5](#)
 - editing, [12-5](#)
- theme styles (*continued*)
 - editing from User Interface page, [12-5](#)
 - Theme Roller, [12-6](#)
 - using, [12-1](#)
- theme subscriptions
 - about, [12-18](#)
 - modifying, [12-18](#)
 - re-subscribing, [12-20](#)
 - refreshing, [12-19](#)
 - unsubscribing, [12-20](#)
 - verifying refresh, [12-19](#)
- themes
 - about, [12-2](#)
 - about subscriptions, [12-18](#)
 - about Universal Theme, [12-2](#)
 - accessing, [12-10](#)
 - changing ID, [12-23](#)
 - changing identification number (ID), [12-23](#)
 - copying, [12-24](#)
 - creating, [12-21](#)
 - creating master application, [12-25](#)
 - default templates, [12-11](#)
 - deleting, [12-24](#)
 - editing, [12-11](#)
 - editing theme attributes, [12-11](#)
 - exporting, [12-28](#), [24-34](#)
 - exporting and importing, [12-28](#)
 - exporting with different ID, [12-23](#)
 - importing, [12-28](#), [24-41](#)
 - live template options, [12-40](#)
 - managing, [12-1](#)
 - reports, [12-29](#)
 - selecting page-level template, [12-34](#)
 - switching, [12-22](#)
 - switching to Universal Theme, [12-2](#)
 - theme styles, [12-4](#)
 - Themes page, [12-10](#)
 - using, [12-1](#)
- Themes page
 - accessing from Shared Components, [12-10](#)
- timeline
 - creating with Create App Wizard, [6-6](#)
- To Do reports
 - Progress Log, [4-19](#)
- To Dos
 - attaching files, [4-17](#)
 - creating, [4-16](#)
 - managing, [4-15](#)
 - To Dos page, [4-16](#)
 - updating a task, [4-17](#)
 - viewing, [4-16](#)
 - viewing reports, [4-18](#)
- To Dos reports
 - Calendar page, [4-18](#)
- translatable messages, defining, [21-17](#)

translating, [21-38](#)
 messages in PL/SQL procedures, [21-17](#)

translation, [21-2](#)
 applications, [21-2](#)
 dynamic, [21-61](#)
 dynamic text strings, [21-4](#)
 dynamic translations, [21-5](#)
 editing manually, [21-15](#)
 exporting text, [21-9](#)
 globalization support, [21-2](#)
 language identification, [21-3](#)
 mapping primary application ID, [21-9](#)
 mapping target application ID, [21-9](#)
 messages, [21-4](#), [21-5](#)
 region titles, [21-5](#)
 rules, [21-3](#)
 seeding, [21-9](#)
 shortcuts, [21-4](#)
 steps, [21-8](#)
 templates, [21-6](#)
 translation file, [21-10](#)
 understanding, [21-8](#)
 understanding application rendering, [21-4](#)
 XLIFF, [21-10](#)
 XLIFF Target Elements, [21-12](#)

translation file, [21-10](#)
 publishing, [21-13](#)
 uploading, [21-13](#)

tree
 creating with Create Page Wizard, [7-6](#), [7-9](#)

trees
 about, [14-38](#)
 creating, [14-41](#)
 creating on new page, [14-39](#)
 editing attributes, [14-44](#)
 managing, [14-38](#)

U

Universal Theme

Migration Guide, [12-2](#)
 Mobile Theme - 51, [12-2](#)
 responsive design, about, [12-2](#)
 switching to, [12-2](#)
 Universal Theme - 42, [12-2](#)

unlocking, productivity application, [3-6](#)

unsaved changes
 warning users, [8-36](#), [14-17](#), [14-35](#)

Upgrade Application icon, [6-74](#)

URL
 publishing, [24-23](#), [24-24](#)
 SSL-enabled, [13-14](#)
 syntax, [2-13](#)

URL tampering
 preventing, [20-30](#)

user identity
 establishing, [20-56](#)
 verifying, [2-9](#)

user interface
 about, [1-2](#), [1-3](#)
 controlling, [12-1](#)

User Interface
 adding, [6-53](#)

User Interface attributes
 accessing, [6-49](#)
 Add New User Interface, [6-50](#)
 configuring, [6-48](#)
 CSS file URLs, [6-52](#)
 Favicon, [6-52](#)
 Favicon HTML, [6-52](#)
 Image Prefix, [6-50](#)
 Logo, [6-51](#)
 Logo Attributes, [6-51](#)
 Logo Type, [6-51](#)
 Media Type, [6-51](#)
 Static File Prefix, [6-50](#)
 User Interface Detection, [6-52](#)
 User Interfaces, [6-50](#)

User Interface Defaults
 exporting, [24-35](#)
 importing, [24-42](#)

User Interface Details, [6-55](#)
 Add "Built with APEX" to Footer, [6-56](#)
 Auto Detect, [6-56](#)
 Concatenated Files, [6-59](#)
 Content Delivery Network, [6-58](#)
 Default, [6-56](#)
 Display Name, [6-55](#)
 Display Navigation, [6-56](#)
 editing, [6-54](#)
 Enable End Users to choose Theme Style, [6-56](#)
 File URLs, [6-58](#), [6-59](#)
 Global Page, [6-56](#)
 Home URL, [6-56](#)
 Include deprecated JavaScript, [6-58](#)
 Include desupported JavaScript, [6-58](#)
 Include jQuery Migrate, [6-59](#)
 Implementation, [6-57](#)
 List Template, [6-57](#)
 Login URL, [6-56](#)
 Navigation Bar List, [6-57](#)
 Navigation Menu List, [6-56](#)
 Position, [6-57](#)
 Sequence, [6-55](#)
 Template Options, [6-57](#)
 Theme, [6-56](#)
 Theme Style, [6-56](#)
 User Interface Type, [6-55](#)

user preferences
 default schema, [1-28](#)
 editing, [1-28](#)
 enabling legacy Component View, [1-28](#)
 running in new window, [1-28](#)
 using single window, [1-28](#)

user roles
 developer, [20-3](#)
 Oracle Application Express administrator,
[20-3](#)
 Workspace administrator, [20-3](#)

users
 about managing, [20-15](#)

Utilities icon, [1-24](#)

V

V function, [2-18](#)

validations
 about, [16-6](#)
 associating with item, [10-22](#)
 creating, [16-7](#), [16-8](#)
 creating with error messages, [10-21](#)
 defining error messages, [16-9](#)
 Go to Error link, [16-11](#)
 making conditional, [16-10](#)
 When Button Pressed attribute, [2-43](#)

Version attribute, [6-28](#)

W

wallets, creating, [20-14](#)

Warn on Unsaved Changes, [8-36](#), [14-17](#)

Web credentials
 creating, [18-37](#)
 deleting unused, [18-39](#)
 editing, [18-38](#)
 viewing history, [18-40](#)
 viewing utilization, [18-39](#)

Web service process
 editing a process, [18-58](#)
 mapping input parameters to static values,
[18-58](#)

Web service reference, [18-42](#)
 creating, [18-43](#)
 creating a report, [18-54](#)
 creating forms, [18-52](#)
 creating forms and reports, [18-51](#)
 creating manually, [18-45](#), [18-46](#)
 specifying WSDL, [18-43](#)
 viewing a history, [18-59](#)

Web Service Reference page, [18-42](#)

Web services, [18-40](#)
 creating a reference from a WSDL, [18-44](#)
 creating RESTful Web services, [18-47](#)

Web services (*continued*)
 editing a Web service process, [18-58](#)
 invoking as a process, [18-56](#)
 specifying proxy server address, [18-42](#)
 SSL enabled, [18-43](#)
 testing a RESTful Web service reference,
[18-49](#)
 testing Web service reference manually,
[18-46](#)
 using, [18-42](#)
 viewing Web service reference history, [18-59](#)
 Web service reference repository, [18-50](#)
 Web Services References page, [18-42](#)

Web Source Modules
 about, [18-27](#)
 copying, [18-32](#)
 creating, [18-28](#)
 deleting, [18-30](#)
 deleting unused, [18-32](#)
 editing, [18-30](#)
 exporting, [18-27](#)
 managing, [18-27](#), [18-33](#)
 viewing history, [18-33](#)
 viewing utilization, [18-32](#)

Websheet applications
 about, [5-1](#)
 access control list, [5-13](#)
 controlling access, [5-13](#)
 copying a page, [5-4](#)
 creating, [5-1](#)
 creating access control list entries, [5-14](#)
 creating from App Builder, [5-2](#)
 creating from running Websheet, [5-3](#)
 deleting, [5-11](#)
 editing page details, [5-6](#)
 monitoring activity, [5-12](#)
 running, [5-4](#)
 specifying access control list, [5-13](#)
 understanding Websheet properties, [5-7](#)
 viewing Websheet dashboard, [5-11](#)

Websheet authentication
 Application Express Account, [5-9](#)
 Custom, [5-9](#)
 LDAP, [5-9](#)
 Single Sign-On, [5-9](#)

Websheet dashboard, viewing, [5-11](#)

Websheet properties
 about, [5-7](#)
 Access Control List Type, [5-10](#)
 Allow Public Access, [5-10](#)
 Authentication, [5-9](#)
 Cookie Domain, [5-9](#)
 Cookie Name, [5-9](#)
 Cookie Path, [5-9](#)
 Default Application Language, [5-8](#)

- Websheet properties (*continued*)
 - Default Application Territory, [5-8](#)
 - Description, [5-8](#)
 - editing, [5-7](#)
 - editing SQL access, [5-11](#)
 - Home Page, [5-8](#)
 - Login Message, [5-8](#)
 - Logo, [5-9](#)
 - Logo Type, [5-9](#)
 - Logout URL, [5-9](#)
 - Name, [5-8](#)
 - Owner, [5-8](#)
 - Secure, [5-9](#)
 - Show Reset Password, [5-8](#)
 - Status, [5-8](#)
 - Style, [5-9](#)
 - Websheet Email From Address, [5-8](#)
 - Websheets
 - about navigation, [3-25](#)
 - editing, [3-35](#)
 - modifying, [3-35](#)
 - navigating, [3-23](#)
 - understanding navigation, [3-25](#)
 - using breadcrumbs, [3-23](#)
 - wizard
 - creating with Create App Wizard, [6-6](#)
 - creating with Create Page Wizard, [7-6](#), [7-9](#)
 - wizards
 - Create Page, [7-1](#)
 - creating a calendar, [11-6](#)
 - creating a chart, [11-29](#)
 - creating a list, [17-42](#), [17-48](#)
 - creating a LOV, [17-12](#)
 - creating a page computation, [16-2](#)
 - creating applications, [2-2](#)
 - creating navigation bar entry, [11-45](#), [15-20](#)
 - Workspace Activity reports
 - Application Errors, [22-5](#)
 - Developer Activity, [22-5](#)
 - Environment, [22-5](#)
 - Login Attempts, [22-5](#)
 - Page View Analysis, [22-5](#)
 - Page Views, [22-5](#)
 - Sessions, [22-5](#)
 - Workspace Schema Reports, [22-5](#)
 - Workspace administrator, [20-3](#)
 - Workspace home page
 - Administration menu, [1-10](#)
 - Workspace home page (*continued*)
 - Available Updates region, [1-13](#)
 - News entries, [1-6](#)
 - system message, [1-6](#)
 - Top Apps region, [1-6](#)
 - Top Users region, [1-6](#)
 - workspace news
 - managing, [4-35](#)
 - workspace provisioning, enabling self service, [20-13](#)
 - workspace settings, configuring, [20-14](#)
 - workspace themes
 - adding, [12-26](#)
 - deleting, [12-27](#)
 - exporting, [12-28](#)
 - managing, [12-26](#), [12-28](#)
 - modifying, [12-28](#)
 - viewing existing, [12-27](#)
 - workspace to schema assignments, managing, [20-15](#)
 - Workspace Utilities
 - App Builder Defaults, [1-20](#)
 - Application Express Views, [1-21](#)
 - Credentials, [1-21](#)
 - Cross Application Reports, [1-21](#)
 - Export, [1-21](#)
 - Manage News, [1-20](#)
 - Remote Servers, [1-20](#)
 - REST Enabled SQL Services, [1-20](#)
 - Workspace Themes, [1-21](#)
 - workspaces, managing, [20-15](#)
 - WSDL document, [18-42](#)
- ## X
-
- XLIFF, [21-10](#)
 - applying, [21-13](#)
 - deleting an uploaded document, [21-14](#)
 - Target Elements, [21-12](#)
 - uploading, [21-13](#)
 - uploading and publishing, [21-13](#)
- ## Z
-
- Zip file
 - downloading static application files, [17-32](#)
 - downloading static workspace files, [17-35](#)