

Oracle® Application Express

App Builder User's Guide



Release 5.1

E64912-05

October 2017

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 5.1

E64912-05

Copyright © 2003, 2017, 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	xliii
Documentation Accessibility	xliii
Related Documents	xliii
Conventions	xliv

Changes in This Release

Changes in Oracle Application Express Release 5.1	xliv
---	------

1 Quick Start

1.1	About Accessing Your Development Environment	1-1
1.2	Understanding Oracle Application Express	1-2
1.2.1	What is Oracle Application Express?	1-2
1.2.2	How Oracle Application Express Works	1-2
1.2.3	About the Differences Between Runtime and Full Development Environments	1-3
1.3	Understanding the Workspace Home Page	1-3
1.3.1	Workspace Home Page	1-4
1.3.2	Top Apps and Top Users Dashboards	1-5
1.3.3	News and Messages	1-5
1.3.4	Developer Navigation Tools	1-6
1.3.4.1	Global Search Field	1-7
1.3.4.2	Administration Menu	1-8
1.3.4.3	Account Menu	1-10
1.3.5	Available Updates Region	1-11
1.3.6	Accessibility Mode Region	1-11
1.4	Understanding App Builder	1-11
1.4.1	Accessing App Builder	1-12
1.4.2	Using the App Builder Home Page	1-13
1.4.2.1	App Builder Home Page	1-13
1.4.2.2	Reset and Create Buttons	1-15

1.4.2.3	App Builder Home Recent List	1-16
1.4.2.4	Migrations Link	1-16
1.4.2.5	Using Workspace Utilities	1-16
1.4.3	Accessing the Application Home Page	1-16
1.4.4	Using the Application Home Page	1-17
1.4.4.1	Edit Application Properties Button	1-17
1.4.4.2	Application Home Page Icons	1-18
1.4.4.3	Create Page Button	1-19
1.4.4.4	Application Home Page Navigation Bar	1-20
1.4.4.5	About Customizing the Application Home Page	1-21
1.4.4.6	Application Home Page Tasks List	1-21
1.4.4.7	Recently Edited Pages	1-22
1.5	Editing Your Account Profile	1-22
1.5.1	Editing Your Account Profile	1-22
1.5.2	Changing Your Password	1-23
1.6	Editing User Preferences	1-23
1.7	Using Oracle Application Express Documentation	1-25
1.7.1	Accessing Oracle Help Center	1-25
1.7.2	Searching the Oracle Help Center	1-26
1.7.3	Downloading the Documentation Library	1-26
1.7.4	Viewing Field-Level Help	1-27
1.7.4.1	Viewing Field-level Help in Oracle Application Express	1-27
1.7.4.2	Viewing Help in Page Designer	1-28

2 App Builder Concepts

2.1	Understanding Oracle Application Express Applications	2-1
2.1.1	What is an Oracle Application Express Application?	2-2
2.1.2	What Is an Application Page?	2-2
2.1.3	What is the Difference Between a Database and Websheet Application?	2-2
2.1.3.1	About Database Applications	2-2
2.1.3.2	About Websheet Applications	2-3
2.1.4	About Running the Create Application Wizard	2-3
2.2	Managing User Interface Design	2-5
2.2.1	About Determining Your Application User Interface	2-5
2.2.2	About Mobile Application Design	2-5
2.2.3	Understanding the Difference Between Mobile Design and Responsive Design	2-6
2.3	Running an Application or Page	2-7
2.3.1	Run Application and Run Page Buttons	2-7
2.3.2	How Your Browser Impacts the Way Applications Run	2-8

2.3.3	Running an Application from the App Builder Home Page	2-8
2.3.4	Running an Application from the Application Home Page	2-8
2.3.5	Running a Page on the Application Home Page	2-9
2.4	How Does Page Processing and Page Rendering Work?	2-9
2.4.1	How the Application Express Engine Renders and Processes Pages	2-10
2.4.2	About Implicit Commit Points	2-10
2.4.3	Understanding Conditional Rendering and Processing	2-10
2.4.3.1	What is a Condition?	2-11
2.4.3.2	Selecting a Condition	2-11
2.4.3.3	Condition Example: Current Page in Expression 1	2-11
2.4.3.4	Condition Example: Exists	2-11
2.4.3.5	Condition Example: PL/SQL Expression	2-12
2.4.4	About Verifying User Identity	2-12
2.4.5	About Controlling Access to Controls and Components	2-12
2.5	About Enabling Support for Bookmarks	2-12
2.6	Understanding Session State Management	2-13
2.6.1	What is Session State?	2-13
2.6.2	What Is a Session?	2-13
2.6.3	About Session IDs	2-14
2.6.4	Viewing Session State	2-14
2.6.4.1	Viewing Session State for a Page	2-14
2.6.4.2	Using the Session Page	2-15
2.7	Understanding URL Syntax	2-16
2.7.1	About the Application Page URL	2-16
2.7.2	About Using f?p Syntax to Link Pages	2-17
2.7.3	Application Alias	2-19
2.7.4	Page Alias	2-19
2.7.5	Calling a Page Using an Application and Page Alias	2-19
2.7.6	Calling a Page from a Button URL	2-20
2.8	Managing Session State Values	2-20
2.8.1	About Referencing Session State	2-20
2.8.2	Setting Session State	2-21
2.8.2.1	About Setting Session State	2-21
2.8.2.2	About Setting Session State with a Form Submission	2-22
2.8.3	Clearing Session State	2-22
2.8.3.1	About Clearing Cache for an Item	2-22
2.8.3.2	Clearing Cache for All Page Items	2-22
2.8.3.3	Example: Clearing Cache for an Entire Application	2-24
2.8.3.4	About Resetting an Application Completely	2-24
2.8.3.5	About Clearing Cache for the Current User Session	2-24
2.8.4	Referencing Session State Using Bind Variable Syntax	2-24

2.8.4.1	About Using Bind Variable Syntax	2-24
2.8.4.2	About Using Bind Variables in Regions Based on a SQL Query or LOV	2-25
2.8.4.3	About Using Bind Variables in Regions Based on PL/SQL	2-25
2.8.5	About Session Cloning	2-25
2.9	Understanding Substitution Strings	2-26
2.9.1	About Using Substitution Strings	2-26
2.9.2	Controlling Output Escaping in Substitution Strings	2-27
2.9.3	About Determining Substitution String Usage within Templates	2-28
2.9.4	Using Built-in Substitution Strings	2-28
2.9.4.1	APEX\$ROW_NUM	2-29
2.9.4.2	APEX\$ROW_SELECTOR	2-30
2.9.4.3	APEX\$ROW_STATUS	2-30
2.9.4.4	APP_ID	2-30
2.9.4.5	APP_ALIAS	2-31
2.9.4.6	APP_AJAX_X01, ... APP_AJAX_X10	2-31
2.9.4.7	APP_DATE_TIME_FORMAT	2-32
2.9.4.8	APP_IMAGES	2-32
2.9.4.9	APP-NLS_DATE_FORMAT	2-33
2.9.4.10	APP-NLS_TIMESTAMP_FORMAT	2-33
2.9.4.11	APP-NLS_TIMESTAMP_TZ_FORMAT	2-34
2.9.4.12	APP_PAGE_ALIAS	2-34
2.9.4.13	APP_PAGE_ID	2-34
2.9.4.14	APP_REQUEST_DATA_HASH	2-35
2.9.4.15	APP_SESSION	2-35
2.9.4.16	APP_SESSION_VISIBLE	2-36
2.9.4.17	APP_TITLE	2-36
2.9.4.18	APP_UNIQUE_PAGE_ID	2-37
2.9.4.19	APP_USER	2-37
2.9.4.20	AUTHENTICATED_URL_PREFIX	2-38
2.9.4.21	BROWSER_LANGUAGE	2-39
2.9.4.22	CURRENT_PARENT_TAB_TEXT	2-39
2.9.4.23	DEBUG	2-39
2.9.4.24	HOME_LINK	2-40
2.9.4.25	IMAGE_PREFIX	2-40
2.9.4.26	JET_BASE_DIRECTORY	2-41
2.9.4.27	JET_CSS_DIRECTORY	2-41
2.9.4.28	JET_JS_DIRECTORY	2-41
2.9.4.29	LOGIN_URL	2-41
2.9.4.30	LOGOUT_URL	2-42
2.9.4.31	PRINTER_FRIENDLY	2-42

2.9.4.32	PROXY_SERVER	2-42
2.9.4.33	PUBLIC_URL_PREFIX	2-43
2.9.4.34	REQUEST	2-43
2.9.4.35	Using REQUEST	2-43
2.9.4.36	SCHEMA_OWNER	2-45
2.9.4.37	SQLERRM	2-45
2.9.4.38	SYSDATE_YYYYMMDD	2-45
2.9.4.39	THEME_DB_IMAGES	2-46
2.9.4.40	THEME_IMAGES	2-46
2.9.4.41	WORKSPACE_IMAGES	2-46
2.9.4.42	WORKSPACE_ID	2-47

3 Utilizing Packaged Applications

3.1	About Packaged Applications	3-1
3.2	Managing Packaged Applications in a Full Development Environment	3-2
3.2.1	Viewing Packaged Apps	3-3
3.2.2	Installing a Packaged Application	3-4
3.2.3	Running an Installed Packaged Application	3-5
3.2.4	Removing an Installed Packaged Application	3-6
3.2.5	Changing Authentication for an Installed Packaged Application	3-6
3.2.6	Unlocking an Installed Productivity Application	3-7
3.2.7	Updating a Packaged Application	3-7
3.3	Understanding Sample Database Application	3-8
3.3.1	Installing and Running Sample Database Application	3-8
3.3.2	Understanding Sample Database Application	3-9
3.3.2.1	Home Page	3-10
3.3.2.2	Customers Page	3-11
3.3.2.3	Using the Customers Page	3-11
3.3.2.4	Products Page	3-12
3.3.2.5	Using the Products Page	3-13
3.3.2.6	Orders Page	3-14
3.3.2.7	Using the Orders Page	3-14
3.3.2.8	Reports Page	3-15
3.3.2.9	Administration Page	3-15
3.3.2.10	Accessing the Mobile Version	3-16
3.3.3	Modifying Sample Database Application	3-17
3.3.3.1	Why Modify a Sample Database Application?	3-18
3.3.3.2	Editing Sample Database Application	3-18
3.4	Understanding AnyCo IT Department Websheet	3-19
3.4.1	Installing and Running AnyCo IT Department	3-19

3.4.2	Understanding Websheet Navigation	3-20
3.4.3	Understanding AnyCo IT Department	3-21
3.4.3.1	Navigating Between Pages and Viewing Page Sections	3-22
3.4.3.2	Home Page	3-23
3.4.3.3	Projects Page	3-25
3.4.3.4	Project Review Page	3-25
3.4.3.5	Systems Page	3-26
3.4.3.6	Planned Purchase Review	3-27
3.4.3.7	Systems Budget Review	3-28
3.4.4	Modifying Websheet Content	3-28
3.4.4.1	Why Modify Sample Websheet Application?	3-29
3.4.4.2	Updating a Websheet Section	3-29
3.5	Utilizing Packaged App Administration	3-31
3.5.1	Making a Service Request	3-31
3.5.2	Setting Workspace Preferences	3-32
3.5.3	Managing Users	3-33
3.5.4	Managing Saved Interactive Reports	3-33
3.5.5	Managing Interactive Report Subscriptions	3-34
3.5.6	Viewing Activity Reports	3-34
3.5.7	Viewing Page Views	3-34
3.5.8	Viewing Top Users	3-35
3.6	Using a Packaged Apps Only Workspace	3-35
3.6.1	What Is a Packaged Apps Only Workspace?	3-35
3.6.2	Requesting a Packaged Apps Only Workspace	3-35
3.6.3	Signing In To a Packaged Apps Only Workspace	3-36
3.6.4	Using the Packaged App Page	3-37

4 Managing the Application Life Cycle with Team Development

4.1	Accessing Team Development	4-1
4.2	Team Development Home Page	4-2
4.3	About Utilizing the Tags Summary	4-3
4.4	Tracking Features	4-3
4.4.1	Viewing the Features Report	4-4
4.4.2	Creating a Feature	4-4
4.4.3	About Creating Parent Features	4-5
4.4.4	Updating a Feature	4-5
4.4.5	How to Attach Files to a Feature	4-5
4.4.5.1	Attaching a File to a Feature	4-6
4.4.5.2	Viewing, Editing, and Removing Attached Files	4-6
4.4.6	Viewing Other Feature Reports	4-7

4.4.6.1	Viewing the Features Tree	4-7
4.4.6.2	Viewing the Features Calendar	4-8
4.4.6.3	Viewing The Feature History Report	4-8
4.4.6.4	Viewing the Feature Progress Log	4-8
4.4.6.5	Viewing the Feature Focus Areas Chart	4-9
4.4.6.6	Viewing the Approval Statuses	4-9
4.4.6.7	Viewing the Feature Owners Chart	4-9
4.4.6.8	Feature Utilities	4-10
4.5	Tracking Milestones	4-10
4.5.1	Viewing the Milestones Report	4-10
4.5.2	Creating a Milestone	4-11
4.5.3	Updating a Milestone	4-11
4.5.4	Viewing Other Milestone Reports	4-11
4.5.4.1	Viewing the Milestone Calendar	4-12
4.5.4.2	Viewing the Milestones By Owner Report	4-12
4.5.4.3	Viewing the Features by Milestone Report	4-12
4.6	Tracking To Dos	4-13
4.6.1	Viewing To Dos	4-13
4.6.2	Creating a To Do	4-13
4.6.3	Updating a To Do	4-14
4.6.4	Managing Files Attached to a To Do	4-14
4.6.4.1	Attaching a File to a To Do	4-15
4.6.4.2	Viewing, Editing, and Removing Attached Files	4-15
4.6.5	Viewing Other To Do Reports	4-15
4.6.5.1	Viewing the To Do Calendar	4-16
4.6.5.2	Viewing the To Do Progress Log	4-16
4.7	Managing Bugs	4-16
4.7.1	Viewing and Editing Bugs	4-17
4.7.2	Creating a Bug	4-17
4.7.3	Editing a Bug	4-17
4.7.4	Managing Files Attached to a Bug	4-18
4.7.4.1	Attaching a File to a Bug	4-18
4.7.4.2	Viewing, Editing, and Removing Attached Files	4-19
4.7.5	Viewing Other Bug Reports	4-19
4.7.5.1	Bugs Dashboard	4-19
4.7.5.2	Viewing the Bug Calendar	4-20
4.7.5.3	Viewing the Bugs By Developer	4-20
4.7.5.4	Viewing the Bugs By Day	4-20
4.8	Managing Feedback	4-20
4.8.1	Viewing Feedback	4-21
4.8.2	Adding a Feedback Page to an Application	4-21

4.8.3	Submitting Feedback	4-22
4.8.4	Reviewing Feedback	4-22
4.8.5	Viewing Other Feedback Reports	4-23
4.8.5.1	Viewing the Feedback Calendar	4-23
4.8.5.2	Viewing Feedback By Filing User	4-23
4.9	Managing Team Development Utilities	4-24
4.9.1	Accessing Team Development Utilities	4-24
4.9.2	Configuring Team Development Settings	4-26
4.9.3	Viewing a Release Summary	4-26
4.9.4	Managing Feature Utilities	4-27
4.9.5	Editing Focus Areas	4-27
4.9.6	Viewing File Attachments	4-27
4.9.7	Purging Data	4-28
4.9.8	Managing News Entries	4-28
4.9.9	Managing Links	4-29
4.9.9.1	Adding Links to the Links Page	4-29
4.9.9.2	Editing or Deleting a Link	4-29

5 Creating Websheet Applications

5.1	About Websheet Applications	5-1
5.2	Creating a Websheet Application	5-2
5.2.1	Creating a Websheet from App Builder	5-2
5.2.2	Creating a Websheet Application from a Running Websheet	5-3
5.3	Copying an Existing Websheet Page	5-4
5.4	Running a Websheet	5-4
5.4.1	Running a Websheet from the App Builder Home Page	5-5
5.4.2	Running a Websheet from the Application Home Page	5-5
5.4.3	Determining a Websheet URL	5-6
5.5	Editing Page Details	5-6
5.6	Deleting a Websheet Application	5-7
5.7	Viewing the Websheet Dashboard	5-7
5.8	Monitoring Websheet Activity	5-7
5.9	Understanding Websheet Properties	5-8
5.9.1	Editing Websheet Properties	5-8
5.9.2	Websheet Properties	5-9
5.9.2.1	Websheet	5-9
5.9.2.2	Details	5-10
5.9.2.3	Logo	5-10
5.9.2.4	Style	5-10
5.9.2.5	Authentication	5-11

5.9.2.6	Authorization	5-11
5.9.2.7	SQL	5-12
5.10	Controlling Websheet Access	5-12
5.10.1	How Websheet Access Works	5-13
5.10.2	Specifying the Access Control List Type	5-13
5.10.3	Creating Access Control List Entries	5-14

6 Creating Database Applications

6.1	Using the Create Application Wizard	6-1
6.1.1	About the Create Application Wizard	6-2
6.1.2	Available Page Types When Running the Create Application Wizard	6-2
6.2	Creating a Database Application	6-4
6.2.1	Creating a Database Application Based on a Table, Query, or Drill-Down Query	6-5
6.2.2	Creating a Desktop Database Application from a Spreadsheet	6-8
6.2.3	Copying a Database Application	6-9
6.2.3.1	Copying a Database Application Using the Create Application Wizard	6-10
6.2.3.2	Copying an Application from the Application Home Page	6-11
6.3	Creating Mobile Applications	6-11
6.3.1	About jQuery Mobile Support	6-11
6.3.2	Creating a Mobile Application	6-12
6.4	Deleting an Application	6-15
6.4.1	Deleting an Application from the Application Home Page	6-15
6.4.2	Deleting an Application from the Edit Application Definition Page	6-15
6.5	Managing Application Attributes	6-16
6.5.1	Editing Application Attributes	6-16
6.5.1.1	Accessing the Edit Application Definition Page	6-16
6.5.1.2	Edit Application Definition Page	6-17
6.5.2	Configuring Security Attributes	6-23
6.5.2.1	Accessing the Edit Security Attributes Page	6-24
6.5.2.2	Security Attributes Page	6-24
6.5.3	Configuring Globalization Attributes	6-32
6.5.3.1	Accessing the Globalization Attributes Page	6-32
6.5.3.2	Edit Globalization Attributes Page	6-33
6.5.4	Managing the Application User Interface	6-36
6.5.4.1	Accessing User Interface Attributes	6-36
6.5.4.2	User Interface Page	6-37
6.5.4.3	Defining an Application Logo	6-39
6.5.4.4	Creating a New User Interface	6-40
6.5.4.5	User Interface Details Page	6-40

6.6	Managing Pages in a Database Application	6-44
6.6.1	About Adding Pages to a Database Application	6-45
6.6.2	Page Types When Running the Create Page Wizard	6-45
6.6.3	Creating a Page from the Application Home Page	6-48
6.6.4	Creating Dialog Pages	6-49
6.6.4.1	About Supported Dialogs	6-49
6.6.4.2	Using a Dialog Page Template	6-49
6.6.4.3	Defining a Page as a Dialog in Component View	6-50
6.6.4.4	Defining a Page as a Dialog in Page Designer	6-51
6.6.4.5	Generating a Cancel Dialog Dynamic Action on the Dialog Page	6-52
6.6.4.6	Generating a Close Dialog Page Process on the Dialog Page	6-52
6.6.4.7	About Branching with Dialog Pages	6-53
6.6.4.8	Handling Refresh Actions on Parent Page	6-53
6.6.5	About Copying Database Application Pages	6-54
6.6.6	About Copying Pages Between User Interfaces	6-54
6.6.7	About Deleting Database Application Pages	6-55
6.7	Using the Find Icon	6-55
6.7.1	Using the Find Icon	6-56
6.7.2	Searching for Items	6-56
6.7.3	Searching for Pages	6-57
6.7.4	Searching for Queries	6-58
6.7.5	Searching for Tables	6-59
6.7.6	Using the PL/SQL Finder	6-59
6.7.7	Searching for Images	6-60
6.8	Creating Page Groups	6-61
6.8.1	Viewing Page Groups	6-61
6.8.2	Creating a Page Group	6-62
6.8.3	Editing a Page Group Definition	6-62
6.8.4	Assigning a Page to a Page Group	6-63
6.8.5	Viewing Pages by Page Group	6-63
6.8.6	Deleting a Page Group	6-63
6.9	Locking and Unlocking a Page	6-64
6.9.1	Determining If a Page Is Locked	6-64
6.9.2	Locking a Page	6-64
6.9.2.1	Locking a Page from the Application Home Page	6-65
6.9.2.2	Locking a Page from the Page Locks Page	6-65
6.9.2.3	Locking a Page in Page Designer	6-65
6.9.2.4	Locking a Page in Legacy Component View	6-66
6.9.3	Unlocking a Page	6-66
6.9.3.1	Unlocking Pages from Utilities	6-66
6.9.3.2	Unlocking Pages from the Application Home Page	6-67

6.9.3.3	Unlocking a Page in Page Designer	6-67
6.9.3.4	Unlocking a Page in Legacy Component View	6-67
6.9.3.5	Unlocking Pages As a Workspace Administrator	6-67
6.10	Adding Developer Comments	6-68
6.10.1	Developer Comment, Bug, or To Do Button	6-68
6.10.2	Adding Developer Comments to an Application	6-69
6.10.3	Viewing and Editing Developer Comments	6-69
6.10.4	Deleting Developer Comments	6-69
6.10.4.1	Deleting a Specific Developer Comment	6-69
6.10.4.2	Deleting Multiple Developer Comments	6-70
6.10.5	Accessing the Developer Comments from Shared Components	6-70
6.10.6	Viewing the Developer Comments Calendar	6-71
6.11	Adding Database Application Comments	6-71
6.12	Managing Application Groups	6-71
6.12.1	Viewing Database Application Groups	6-72
6.12.2	Creating a Database Application Group	6-72
6.12.3	Assigning a Database Application to an Application Group	6-72
6.12.4	Removing a Database Application from an Application Group	6-73
6.12.5	Deleting an Database Application Group	6-73
6.13	Using Application Utilities	6-74
6.13.1	Accessing the Utilities Page	6-74
6.13.2	Viewing the Application Dashboard	6-75
6.13.3	Upgrading an Application to Include New Components	6-76
6.13.3.1	About Upgrading to Include New Components	6-76
6.13.3.2	Running Upgrade Application	6-77
6.13.4	Viewing the Recently Updated Pages Report	6-77
6.13.5	Viewing the Change History Report	6-77
6.13.6	Using the Database Object Dependencies Report	6-78
6.13.7	Accessing Application Express Views	6-78
6.13.8	Accessing Page Specific Utilities	6-79
6.13.9	Accessing Cross Page Utilities	6-80
6.13.10	Accessing Cross Application Reports	6-80
6.13.11	Using Interactive Report Utilities	6-81
6.13.11.1	Deleting Saved Interactive Reports	6-81
6.13.11.2	Viewing Subscriptions	6-82

7 Managing Pages in Page Designer

7.1	About Page Designer	7-1
7.2	Understanding Page Designer UI Elements	7-3
7.2.1	Page Designer Toolbar	7-3

7.2.2	Left Pane of Page Designer	7-7
7.2.3	Central Pane of Page Designer	7-11
7.2.3.1	Layout Tab	7-11
7.2.3.2	Gallery	7-13
7.2.3.3	Component View Tab	7-14
7.2.3.4	Messages Tab	7-15
7.2.3.5	Page Search Tab	7-16
7.2.3.6	Help Tab	7-17
7.2.4	Right Pane of Page Designer (Property Editor)	7-18
7.2.5	Code Editor	7-20
7.3	Using Page Designer	7-21
7.3.1	Viewing a Page in Page Designer	7-22
7.3.2	Searching for Pages	7-23
7.3.3	Switching Between Three Pane and Two Pane Mode	7-23
7.3.4	Accessing Property Editor	7-24
7.3.5	Adding a Region, Item, or Button by Dragging and Dropping	7-25
7.3.6	About Adding Components Using Gallery Context Menus	7-26
7.3.7	Adding a Region, Item, or Button from the Rendering Tab	7-27
7.3.8	Adding a Region, Item, or Button from the Component View Tab	7-28
7.3.9	Copying a Region, Item, or Button	7-29
7.3.10	Creating Pre-Rendering Branches, Computations, and Processes	7-30
7.3.11	Creating Page Submission Branches, Validations, Computations, and Processes	7-30
7.4	Managing Page Attributes in Page Designer	7-31
7.4.1	Editing Page Attributes in the Property Editor	7-32
7.4.2	Incorporating JavaScript Using Page Designer	7-33
7.4.3	Configuring Rejoin Sessions in Page Designer	7-34
7.4.4	Configuring Dialog Attributes in Page Designer	7-35
7.4.5	Configuring Page Attributes to Warn Users of Unsaved Changes	7-36
7.5	Running a Page from Page Designer	7-37
7.6	Runtime Developer Toolbar	7-38
7.7	Copying a Database Application Page in Page Designer	7-39
7.8	Deleting Pages in Page Designer	7-40
7.8.1	Deleting a Page in Page Designer	7-40
7.8.2	Deleting Multiple Pages	7-41
7.8.3	Deleting a Range of Pages	7-41
7.9	Switching from Legacy Component View to Page Designer	7-41

8 Developing Reports

8.1	Understanding Report Types	8-1
8.1.1	Interactive Grid	8-1

8.1.2	Interactive Reports	8-2
8.1.3	Classic Report	8-4
8.1.4	Report and Form	8-4
8.2	Creating a Report	8-5
8.2.1	Creating Reports Using a Wizard	8-5
8.2.1.1	Report Options When Running the Create Application Wizard	8-6
8.2.1.2	Report Options When Running the Create Page Wizard	8-7
8.2.1.3	Creating a Report Using the Create Application Wizard	8-9
8.2.1.4	Creating a Report Using the Create Page Wizard	8-11
8.2.2	Creating a Report by Dragging and Dropping from the Gallery	8-11
8.3	Managing Interactive Grids	8-13
8.3.1	Managing Interactive Grid Region Attributes	8-14
8.3.1.1	Editing Interactive Grid Region Attributes	8-14
8.3.1.2	Creating Master Detail from an Existing Interactive Grid	8-15
8.3.1.3	Displaying Interactive Grids Conditionally	8-19
8.3.2	Managing Interactive Grid Attributes	8-20
8.3.2.1	Editing Interactive Grid Attributes	8-20
8.3.2.2	Making an Existing Interactive Grid Editable	8-21
8.3.2.3	Controlling Interactive Grid Pagination	8-22
8.3.2.4	Customizing the Interactive Grid Toolbar	8-23
8.3.2.5	Enabling Users to Save Public Reports in Interactive Grids	8-24
8.3.2.6	Restricting Who Can Save Public Reports	8-25
8.3.2.7	Enabling Icon View in an Interactive Grid	8-26
8.3.2.8	Enabling Detail View in an Interactive Grid	8-28
8.3.3	Managing Interactive Grid Column Attributes	8-29
8.3.3.1	Editing Grid Column Attributes	8-30
8.3.3.2	Defining a Column as a List of Values for Filtering	8-31
8.3.3.3	Creating a Column Link in an Interactive Grid	8-32
8.3.4	About Customizing Interactive Grids in a Running Application	8-33
8.3.5	Saving Interactive Grids	8-33
8.3.5.1	How the User Type Effects Available Save Options	8-34
8.3.5.2	About Configuration Dependencies When Saving Interactive Grids	8-35
8.3.5.3	Saving a Private Interactive Grid	8-35
8.3.5.4	Saving a Public Interactive Grid	8-36
8.3.5.5	Saving or Renaming a Primary Interactive Grid	8-36
8.3.5.6	Saving an Alternative Interactive Grid	8-37
8.3.5.7	Renaming a Saved Interactive Grid Report	8-37
8.3.5.8	Deleting a Saved Interactive Grid Report	8-37
8.3.5.9	Resetting a Saved Interactive Grid Report	8-38
8.4	Managing Interactive Reports	8-38

8.4.1	Editing Interactive Reports in Page Designer	8-38
8.4.1.1	Editing Interactive Report Region Attributes in Page Designer	8-39
8.4.1.2	Editing Interactive Report Column Attributes in Page Designer	8-40
8.4.1.3	About Filtering on Conditional Links in Interactive Report Columns	8-41
8.4.1.4	Editing Interactive Report Attributes in Page Designer	8-42
8.4.1.5	Controlling Interactive Report Pagination in Page Designer	8-43
8.4.1.6	Customizing the Interactive Report Search Bar	8-43
8.4.1.7	Customizing the Actions Menu in Page Designer	8-47
8.4.1.8	Configuring Actions Menu Download Options in Page Designer	8-47
8.4.1.9	Managing Link Columns in Page Designer	8-52
8.4.1.10	Creating a Column Link in an Interactive Report	8-56
8.4.1.11	Displaying Interactive Reports Conditionally	8-57
8.4.1.12	Defining a Column as a List of Values in an Interactive Report	8-57
8.4.1.13	Formatting Columns in an Interactive Report	8-58
8.4.1.14	Enabling Icon View in Page Designer	8-59
8.4.1.15	Enabling Detail View in Page Designer	8-61
8.4.1.16	Configuring Advanced Attributes for Interactive Reports	8-62
8.4.2	Editing Interactive Reports in Legacy Component View	8-64
8.4.2.1	Editing Interactive Report Attributes in Component View	8-65
8.4.2.2	Editing Column Attributes in Component View	8-67
8.4.2.3	Creating Column Groups	8-68
8.4.2.4	Controlling Interactive Report Pagination in Component View	8-68
8.4.2.5	Managing the Search Bar and Actions Menu in Component View	8-69
8.4.2.6	Managing Link Columns in Component View	8-76
8.4.2.7	Creating a Column Link in an Interactive Report	8-79
8.4.2.8	Displaying Interactive Reports Conditionally	8-80
8.4.2.9	Defining a Column as a List of Values in an Interactive Report	8-81
8.4.2.10	Formatting Columns in an Interactive Report	8-81
8.4.2.11	Enabling Icon View in Component View	8-82
8.4.2.12	Enabling Detail View in Component View	8-83
8.4.2.13	Configuring Interactive Report Advanced Attributes	8-84
8.4.2.14	Editing Saved Default Interactive Reports in Component View	8-86
8.4.2.15	Viewing Saved Interactive Report Activity	8-87
8.4.3	Linking to Interactive Reports	8-87
8.4.3.1	Request Syntax	8-88
8.4.3.2	Name and Value Syntax	8-88
8.4.3.3	ClearCache Syntax	8-89
8.4.3.4	Restrictions When Linking to Interactive Reports	8-89
8.4.3.5	Valid Linking Examples	8-90
8.4.3.6	Invalid Linking Examples	8-90
8.4.3.7	Linking to Shared Interactive Reports	8-91

8.4.4	Customizing Interactive Reports in a Running Application	8-92
8.4.4.1	About the Search Bar	8-93
8.4.4.2	About the Actions Menu	8-94
8.4.4.3	Using Interactive Report Filters	8-95
8.4.4.4	Saving Interactive Reports	8-96
8.5	Managing Classic Reports	8-101
8.5.1	Editing Classic Reports in Page Designer	8-102
8.5.1.1	Altering Classic Report Layout in Page Designer	8-102
8.5.1.2	Editing Classic Report Attributes in Page Designer	8-103
8.5.1.3	Editing Classic Report Pagination in Page Designer	8-104
8.5.1.4	Enabling Classic Report CSV Output Option in Page Designer	8-105
8.5.1.5	Adding Support for XML File or a CSV File Export in Page Designer	8-106
8.5.1.6	Enabling Column Sorting in a Classic Report in Page Designer	8-107
8.5.1.7	Creating a Column Link in a Classic Report in Page Designer	8-107
8.5.1.8	Adding a Download Link to a Classic Report in Page Designer	8-108
8.5.1.9	Defining an Updatable Column in a Classic Report in Page Designer	8-109
8.5.1.10	Controlling When Classic Report Columns Display in Page Designer	8-109
8.5.1.11	Defining a Column as a List of Values in a Classic Report in Page Designer	8-110
8.5.1.12	Controlling Classic Report Column Breaks in Page Designer	8-111
8.5.2	Editing Classic Reports in Legacy Component View	8-112
8.5.2.1	Editing Classic Report Attributes in Component View	8-112
8.5.2.2	Altering Classic Report Layout in Component View	8-113
8.5.2.3	Controlling Classic Report Pagination in Component View	8-114
8.5.2.4	Enabling Column Sorting in a Classic Report in Component View	8-117
8.5.2.5	Adding a Download Link to a Classic Report in Component View	8-118
8.5.2.6	Enabling Classic Report CSV Output in Component View	8-118
8.5.2.7	Adding Support for XML File or a CSV File Export	8-118
8.5.2.8	Creating a Column Link in a Classic Report in Component View	8-119
8.5.2.9	Defining an Updatable Column in Component View	8-120
8.5.2.10	Defining a Column as a List of Values in a Classic Report in Component View	8-121
8.5.2.11	Controlling When Classic Report Columns Display in Component View	8-122
8.5.2.12	Controlling Classic Report Column Breaks in Component View	8-123
8.5.2.13	Synchronizing Classic Report Column Order with Region Source	8-123
8.6	Printing Report Regions	8-123
8.6.1	About Printing Reports to PDF	8-124
8.6.2	About Report Printing Configuration Options	8-124
8.6.3	About Classic Report Printing Methods	8-125

8.6.4	Configuring Classic Report Region Print Attributes	8-126
8.6.4.1	Configuring a Classic Report Print Attributes in Page Designer	8-126
8.6.4.2	Configuring a Classic Report Print Attributes in Component View	8-127
8.6.5	Configuring Interactive Report Region Print Attributes	8-128
8.6.5.1	Configuring Interactive Report Regions for Printing in Page Designer	8-128
8.6.5.2	Configuring Interactive Report Regions for Printing in Legacy Component View	8-129
8.6.6	About Report Queries	8-130
8.6.7	About Report Layouts	8-130
8.7	Understanding BLOB Support in Forms and Reports	8-130
8.7.1	About BLOB Support in Reports	8-131
8.7.2	About Providing a Download Link	8-131
8.7.3	About Displaying the BLOB	8-132
8.7.4	About Working With BLOBs Procedurally	8-133

9 Developing Forms

9.1	Understanding Form Types	9-1
9.1.1	Editable Interactive Grid	9-1
9.1.2	Form on Table	9-2
9.1.3	Master Detail Forms	9-3
9.2	Creating a Form	9-5
9.2.1	Form Options When Running the Create Page Wizard	9-6
9.2.2	Creating a Form Using the Create Page Wizard	9-8
9.2.3	Creating a Form Using the Create Application Wizard	9-9
9.2.5	Creating a Form Region in Page Designer	9-10
9.2.4	Creating a Form Manually in Legacy Component View	9-11
9.2.4.1	About Creating Forms Manually in Component View	9-11
9.2.4.2	Processing a Form	9-11
9.2.4.3	Populating Forms in Component View	9-13
9.3	About Making an Interactive Grid Editable	9-14
9.4	Managing Master Detail Forms	9-14
9.4.1	About Master Detail Forms	9-15
9.4.2	Creating a Single Page Master Detail Using the Create Application Wizard	9-18
9.4.3	Creating a Single Page Master Detail Using the Create Page Wizard	9-19
9.4.4	Creating a Two Page Master Detail Using the Create Page Wizard	9-21
9.5	Validating User Input in Forms	9-22
9.5.1	About Creating Validations and Error Messages	9-22
9.5.2	Associating a Validation with a Specific Item in Page Designer	9-22
9.5.3	Associating a Validation with a Specific Item in Component View	9-23

9.5.4	About the Error Message Display Location	9-24
9.6	Understanding BLOB Support in Forms	9-24
9.6.1	About BLOB Support in Forms	9-24
9.6.2	About Uploading and Downloading Files into a BLOB	9-24
9.6.3	About Displaying the BLOB	9-25
9.6.4	About Removing the Image Reference	9-25

10 Managing Database Application Components

10.1	Creating Calendars	10-1
10.1.1	About Supported Calendars	10-2
10.1.2	Sample Calendar	10-3
10.1.3	Adding a Calendar to a New Page	10-4
10.1.4	Creating a Calendar in Page Designer	10-5
10.1.5	Managing Calendar Attributes in Page Designer	10-6
10.1.5.1	Editing Calendar Attributes in the Property Editor	10-7
10.1.5.2	Editing an Existing Calendar to Include Add and Edit Functionality	10-7
10.1.5.3	Adding Calendar Initialization JavaScript Code	10-9
10.1.5.4	Enabling the Dragging and Dropping of Data In an Existing Calendar	10-10
10.1.6	About Dynamic Action Support for Calendar	10-11
10.2	Creating Maps	10-12
10.2.1	About Map Chart Support	10-13
10.2.2	About Creating SQL Queries for Maps	10-13
10.2.3	Adding Map Charts to a New Page	10-16
10.2.4	Managing Map Charts in Page Designer	10-17
10.2.4.1	Adding a Map Chart in Page Designer	10-17
10.2.4.2	Editing Map Chart Attributes in Page Designer	10-18
10.2.4.3	Using Custom XML with Map Charts	10-19
10.2.4.4	Enabling Automatic Updates	10-20
10.2.5	Managing Map Charts in Legacy Component View	10-20
10.2.5.1	Adding a Map Chart in Component View	10-21
10.2.5.2	Editing Map Chart Attributes	10-22
10.2.5.3	Using Custom XML with Map Charts	10-22
10.2.5.4	Enabling Asynchronous Updates	10-23
10.3	Creating Charts	10-23
10.3.1	About Supported Charts	10-24
10.3.2	Sample Charts	10-24
10.3.3	Adding a Chart to a New Page	10-25
10.3.4	Managing Charts	10-27
10.3.4.1	Adding a Chart in Page Designer	10-27

10.3.4.2	Editing Chart Attributes in Page Designer	10-29
10.3.4.3	Switching Chart Type in Page Designer	10-29
10.3.4.4	Enabling Automatic Refresh	10-30
10.3.5	Using Custom JavaScript with Charts	10-32
10.4	Creating Help for Your Application	10-33
10.4.1	Creating Help in Page Designer	10-33
10.4.1.1	About Creating Help	10-33
10.4.1.2	Creating a Help Page and Region in Page Designer	10-33
10.4.1.3	Defining Page Help in Page Designer	10-35
10.4.1.4	Defining Item Help in Page Designer	10-35
10.4.2	Creating Help in Legacy Component View	10-35
10.4.2.1	About Creating Help	10-36
10.4.2.2	Creating a Help Page and Region in Component View	10-36
10.4.2.3	Defining Page Help in Component View	10-37
10.4.2.4	Defining Item Help in Component View	10-37
10.4.3	Editing Multiple Item Help Topics Simultaneously	10-38
10.4.3.1	Using the Bulk Edit Item Help Report	10-38
10.4.3.2	Seeding Item Help Topics	10-38
10.4.4	Creating a Help Navigation Bar Entry	10-39

11 Using Themes and Theme Styles

11.1	Using Themes	11-1
11.1.1	About Themes	11-1
11.1.2	About Responsive Design	11-2
11.1.3	About Grid-based Themes	11-3
11.1.4	Using Theme Styles	11-3
11.1.4.1	About Theme Styles	11-3
11.1.4.2	Creating a Theme Style	11-4
11.1.4.3	Editing a Theme Style from the Themes Page	11-5
11.1.4.4	Editing a Theme Style from the User Interface Page	11-5
11.1.4.5	Enabling Users to Select a Theme Style	11-5
11.1.4.6	Using Theme Roller	11-6
11.1.5	Understanding Template Options	11-8
11.1.5.1	About Template Options	11-8
11.1.5.2	About Default Template Options	11-9
11.1.5.3	About Template Option Groups and Presets	11-9
11.1.6	Accessing the Themes Page	11-9
11.1.7	Editing Themes	11-10
11.1.7.1	Editing a Theme	11-10
11.1.7.2	Theme Attributes	11-11

11.2	Managing Themes and Subscriptions	11-17
11.2.1	Understanding Theme Subscriptions	11-17
11.2.1.1	About Theme Subscriptions	11-18
11.2.1.2	About Modifying a Theme Subscription	11-18
11.2.1.3	Verifying if a Theme Refresh Is Needed	11-18
11.2.1.4	Refreshing a Subscribed Theme	11-19
11.2.1.5	Unsubscribing to a Theme	11-19
11.2.1.6	Re-subscribing to a Built-in Theme	11-20
11.2.2	Creating a Theme	11-20
11.2.3	Switching Themes	11-21
11.2.3.1	About Switching the Active Theme	11-21
11.2.3.2	About Switching to the Universal Theme	11-22
11.2.3.3	Switching an Active Theme	11-22
11.2.4	Changing a Theme Identification Number	11-23
11.2.5	Copying a Theme	11-24
11.2.6	Deleting a Theme	11-24
11.2.7	Creating a Master Theme Application	11-25
11.2.7.1	Creating a Master Theme Application by Creating a New Application	11-25
11.2.7.2	Creating a Master Theme Application from an Existing Application	11-25
11.2.8	Managing Workspace Themes	11-25
11.2.8.1	Adding a Workspace Theme to the Theme Repository	11-26
11.2.8.2	Viewing an Existing Workspace Theme	11-27
11.2.8.3	Deleting a Workspace Theme	11-27
11.2.8.4	Modifying a Workspace Theme	11-27
11.2.8.5	Exporting a Workspace Theme	11-28
11.2.9	Managing Instance Themes	11-28
11.2.10	About Exporting and Importing Themes	11-28
11.2.11	Viewing Theme Reports	11-29
11.2.11.1	Viewing All Templates in a Theme	11-29
11.2.11.2	Viewing Theme Template Counts	11-29
11.2.11.3	Viewing File References	11-30
11.2.11.4	Viewing Class References	11-30
11.2.11.5	Viewing Template Substitution Strings	11-31
11.3	Creating Custom Themes	11-32
11.3.1	About Cascading Style Sheets	11-32
11.3.2	About Calling the JavaScript File from the Page Template	11-33
11.3.3	About Using Escaping Syntax in Substitution Strings	11-33
11.3.4	Selecting a Default Page Template	11-34
11.3.4.1	Selecting a Page-level Template Within a Theme	11-34
11.3.4.2	Selecting a Page-level Template for a Specific Page	11-34

11.3.5	Creating a New Template	11-35
11.3.6	Using Template Options	11-35
11.3.6.1	Managing Global Template Option Groups	11-36
11.3.6.2	Managing Global Template Options	11-37
11.3.6.3	Managing Template Options	11-39
11.3.7	Viewing Template Reports	11-42
11.3.8	Managing Templates	11-43
11.3.8.1	Viewing Templates on the Templates Page	11-43
11.3.8.2	Viewing Templates Associated with a Specific Page	11-44
11.3.8.3	Replacing Templates	11-44
11.3.8.4	Replacing All Templates within an Application	11-45
11.3.8.5	Viewing Region Position Utilization by Page Template	11-45
11.3.8.6	Unsubscribing to Templates	11-46
11.3.8.7	Publishing Templates	11-46
11.3.9	Breadcrumb Templates	11-47
11.3.9.1	About Breadcrumb Style Navigation	11-47
11.3.9.2	Breadcrumb Template Attributes	11-48
11.3.10	Button Templates	11-49
11.3.10.1	Button Template Attributes	11-49
11.3.11	Legacy Calendar Templates	11-51
11.3.11.1	Supported Legacy Calendar Template Substitution Strings	11-52
11.3.11.2	Legacy Calendar Template Attributes	11-52
11.3.12	Label Templates	11-53
11.3.12.1	Label Template Attributes	11-53
11.3.12.2	About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help	11-55
11.3.13	List Templates	11-56
11.3.13.1	About Hierarchical Lists	11-56
11.3.13.2	List Template Attributes	11-57
11.3.14	Page Templates	11-59
11.3.14.1	Page Template Attributes	11-59
11.3.14.2	Required Dialog Page Template Attributes	11-67
11.3.15	Popup LOV Templates	11-67
11.3.15.1	Popup List of Values Template Attributes	11-68
11.3.16	Region Templates	11-69
11.3.16.1	Region Template Attributes	11-70
11.3.17	Report Templates	11-72
11.3.17.1	Generic Column Templates and Named Column Templates	11-72
11.3.17.2	Report Column Template Attributes for Generic Column Templates	11-73
11.3.17.3	Report Column Template Attributes for Named Column Templates	11-77

11.3.17.4	About Using JavaScript in Column Templates	11-80
11.4	Using Custom Cascading Style Sheets	11-81
11.4.1	About Uploading Cascading Style Sheets	11-81
11.4.2	Referencing an Uploaded Cascading Style Sheet in the Page Template	11-81

12 Controlling Page Layout

12.1	About Page Layout in Oracle Application Express	12-1
12.2	Optimizing a Page for Printing	12-2
12.2.1	Selecting a Printer Friendly Template for an Application	12-2
12.2.2	Using f?p Syntax to Toggle to Printer Friendly Mode	12-3
12.3	Creating a Global Page to Display Components on Every Page	12-3
12.3.1	Creating a Global Page	12-3
12.3.2	Navigating to the Global Page	12-4
12.4	About Regions	12-4
12.5	Managing Regions in Page Designer	12-5
12.5.1	Creating a Region in Page Designer	12-5
12.5.2	Managing Regions in Page Designer	12-6
12.5.2.1	Editing Region Attributes in Page Designer	12-7
12.5.2.2	Controlling Region Positioning	12-8
12.5.2.3	Specifying a Region Header and Footer	12-8
12.5.2.4	Enabling Users to Customize a Page	12-8
12.5.2.5	Utilizing Region Caching in Page Designer	12-9
12.5.2.6	Specifying a Static Region ID	12-10
12.5.2.7	Adding a Region Image	12-11
12.5.2.8	Creating a Region Display Selector	12-11
12.5.3	Copying a Region in Page Designer	12-12
12.5.4	Deleting Regions in Page Designer	12-13
12.6	Managing Regions in Legacy Component View	12-13
12.6.1	Region Types When Running the Create Region Wizard	12-13
12.6.2	Creating a Region in Component View	12-18
12.6.3	Managing Regions in Component View	12-18
12.6.3.1	Editing Region Attributes in Component View	12-19
12.6.3.2	How Region Attributes Affect Page Layout	12-19
12.6.3.3	Specifying a Region Header and Footer	12-21
12.6.3.4	Enabling Users to Customize a Page	12-22
12.6.3.5	Utilizing Region Caching in Component View	12-22
12.6.3.6	Specifying a Static Region ID	12-24
12.6.3.7	Adding a Region Image	12-25
12.6.3.8	Editing Multiple Regions at Once in Component View	12-25
12.6.3.9	Managing Subregions in Component View	12-26

12.6.4	Creating a Region Display Selector	12-28
12.6.5	Copying a Region in Component View	12-28
12.6.6	Deleting Regions in Component View	12-29
12.6.6.1	Deleting a Region	12-29
12.6.6.2	Deleting Multiple Regions at Once	12-29
12.7	About Incorporating Content from Other Web Sites	12-30
12.8	About Managing Images	12-31
12.9	Rendering HTML Using Custom PL/SQL	12-31

13 Managing Database Application Controls

13.1	Understanding Page-Level Items	13-1
13.1.1	About the Differences Between Page Items and Application Items	13-2
13.1.2	About Item Naming Conventions	13-2
13.1.3	Referencing Item Values	13-2
13.1.4	About Referencing Items Using JavaScript	13-4
13.1.5	Working with Multiple Select List Item	13-4
13.1.5.1	About Handling Values Returned from a Multiple Select List Item	13-5
13.1.5.2	Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values	13-5
13.1.5.3	Creating a Shuttle Item	13-6
13.2	Managing Page-Level Items in Page Designer	13-8
13.2.1	Viewing Page-Level Items in Page Designer	13-9
13.2.2	Creating Page-Level Items	13-9
13.2.2.1	Creating a Page-Level Item from the Rendering Tab	13-10
13.2.2.2	Adding a Page-Level Item from the Gallery	13-10
13.2.2.3	Creating a Static List of Values	13-11
13.2.2.4	Creating a Cascading List of Values	13-13
13.2.3	Editing Page-Level Items in Page Designer	13-14
13.2.3.1	Editing Page Item Attributes in the Property Editor	13-14
13.2.3.2	Defining Default Values for Page Items	13-15
13.2.3.3	Configuring Page Item Security	13-16
13.2.3.4	Creating a Quick Pick Selection	13-17
13.2.3.5	Displaying Conditional Page Items	13-18
13.2.3.6	Displaying Read Only Page Items	13-18
13.2.3.7	Configuring Item Attributes to Warn Users of Unsaved Changes	13-19
13.2.3.8	Viewing Item Utilities	13-20
13.3	Managing Page-Level Items in Legacy Component View	13-20
13.3.1	Viewing Page-Level Items in the Component View	13-21
13.3.2	Creating Page-Level Items in Component View	13-22
13.3.2.1	Creating a Page-Level Item in Component View	13-22
13.3.2.2	Creating a Static List of Values	13-23

13.3.2.3	Creating a Cascading List of Values	13-25
13.3.2.4	Creating Multiple Items Using a Tabular Form	13-27
13.3.3	Editing Page-Level Items in Component View	13-28
13.3.3.1	Editing Page Item Attributes in Component View	13-28
13.3.3.2	Defining Default Values	13-29
13.3.3.3	Configuring Page Item Security	13-29
13.3.3.4	Creating a Quick Pick Selection	13-31
13.3.3.5	Displaying Conditional Page Items	13-32
13.3.3.6	Displaying Read Only Page Items	13-32
13.3.3.7	Editing Multiple Items Simultaneously	13-33
13.4	Managing Dynamic Actions	13-36
13.4.1	About Dynamic Actions	13-36
13.4.2	Viewing Dynamic Action Examples	13-37
13.4.3	About Dynamic Action Events	13-37
13.4.4	Managing Dynamic Actions in Page Designer	13-40
13.4.4.1	Creating a Dynamic Action in Page Designer	13-40
13.4.4.2	Editing Dynamic Actions in Page Designer	13-43
13.4.4.3	Defining Dynamic Action Event Scope	13-44
13.4.4.4	Deleting a Dynamic Action in Page Designer	13-44
13.4.5	Managing Dynamic Actions in Legacy Component View	13-45
13.4.5.1	Creating a Dynamic Action in Component View	13-45
13.4.5.2	Editing Dynamic Actions in Component View	13-47
13.4.5.3	Adding Additional True Actions	13-48
13.4.5.4	Defining the Frequency and Scope	13-49
13.4.6	About Calling JavaScript Using a Dynamic Action	13-50
13.4.7	Debugging Dynamic Actions	13-50
13.4.7.1	About Debugging Dynamic Actions	13-50
13.4.7.2	Debugging Dynamic Actions	13-50
13.5	Creating Buttons	13-51
13.5.1	What Actions Can a Button Perform?	13-51
13.5.2	Managing Buttons in Page Designer	13-51
13.5.2.1	Creating a Button in Page Designer	13-52
13.5.2.2	Editing a Button in Page Designer	13-53
13.5.2.3	Displaying a Button Conditionally	13-53
13.5.2.4	Configuring Button Attributes to Warn Users of Unsaved Changes	13-54
13.5.3	Managing Buttons in Legacy Component View	13-55
13.5.3.1	Creating a Button in Component View	13-55
13.5.3.2	Creating Multiple Buttons in Component View	13-57
13.5.3.3	Editing a Button in Legacy Component View	13-58
13.5.3.4	Displaying a Button Conditionally	13-59

13.5.4	About Calling JavaScript from a Button	13-60
13.5.5	About the Relationship Between Button Names and REQUEST	13-61
13.5.6	About Branching with Buttons	13-61
13.6	Creating Trees	13-61
13.6.1	About Trees	13-62
13.6.2	About Support for Legacy jsTrees	13-62
13.6.3	Managing Trees in Page Designer	13-62
13.6.3.1	Creating a Tree on New Page	13-63
13.6.3.2	Creating a Tree in Page Designer	13-65
13.6.3.3	Editing Tree Attributes in Page Designer	13-67
13.6.4	Managing Trees in Legacy Component View	13-68
13.6.4.1	Creating a Tree in Component View	13-69
13.6.4.2	Editing Tree Attributes in Component View	13-70
13.7	About Incorporating JavaScript into an Application	13-71

14 Adding Navigation

14.1	About Creating Lists as Navigation	14-1
14.2	Creating Tabs	14-2
14.2.1	About Tabs	14-2
14.2.2	About the Tabs Page	14-3
14.2.3	Checking for Tab Template Support	14-4
14.2.4	Creating a Tab	14-5
14.2.5	Editing Tabs	14-7
14.2.5.1	Editing Multiple Tabs Simultaneously	14-7
14.2.5.2	Managing Standard Tabs	14-7
14.2.6	Accessing Tab Reports	14-8
14.3	Creating Breadcrumbs	14-8
14.3.1	What Are Breadcrumbs?	14-9
14.3.2	Creating a Breadcrumb While Creating a Page	14-9
14.3.3	Creating a Breadcrumb Manually	14-10
14.3.3.1	About Creating a Breadcrumb Manually	14-10
14.3.3.2	Creating Breadcrumbs	14-10
14.3.3.3	Adding Breadcrumb Entries	14-12
14.3.3.4	Adding Breadcrumb Regions	14-13
14.3.3.5	About Creating Dynamic Breadcrumbs	14-16
14.3.4	Editing Breadcrumbs	14-16
14.3.5	Reparenting Breadcrumb Entries	14-17
14.3.6	Deleting Unused Breadcrumb Entries	14-17
14.3.7	Accessing Breadcrumb Reports	14-18
14.4	Creating a Navigation Bar Entry	14-19

14.4.1	About Navigation Bars	14-19
14.4.2	Creating a Navigation Bar Entry	14-20
14.4.2.1	Creating a Navigation Bar Entry from Scratch	14-20
14.4.2.2	Creating a Navigation Bar Entry for Feedback	14-22
14.4.3	Copying a Navigation Bar Entry	14-23
14.4.4	Editing a Navigation Bar Entry	14-23
14.4.5	Editing Multiple Navigation Bar Entries Simultaneously	14-24
14.4.6	Accessing Navigation Bar Entry Reports	14-24
14.5	Controlling Navigation Using Branches	14-25
14.5.1	Creating a Branch in Page Designer	14-25
14.5.2	Creating a Branch in Component View	14-27
14.5.3	About Branching Conditionally	14-28

15 Managing Computations, Validations, and Processes

15.1	Understanding Page Computations	15-1
15.1.1	About Page Computations	15-2
15.1.2	Managing Page Computations in Page Designer	15-2
15.1.2.1	Creating a Page Computation in Page Designer	15-2
15.1.2.2	Editing Page Computation Attributes in Page Designer	15-3
15.1.2.3	About Editing the Computation Point and Source	15-4
15.1.2.4	About Creating Conditional Computations	15-4
15.1.3	Managing Page Computations in Legacy Component View	15-4
15.1.3.1	Creating a Page Computation in Component View	15-5
15.1.3.2	Editing Page Computation Attributes in Component View	15-5
15.1.3.3	About Editing the Computation Point and Source	15-6
15.1.3.4	About Creating Conditional Computations	15-6
15.1.4	Computation Points and Computation Syntax Example	15-6
15.2	Understanding Validations	15-8
15.2.1	About Validations	15-8
15.2.2	Managing Validations in Page Designer	15-8
15.2.2.1	Creating a Validation in Page Designer	15-9
15.2.2.2	Editing a Validation in Page Designer	15-10
15.2.3	Managing Validations in Legacy Component View	15-11
15.2.3.1	Creating a Validation for a Page Item	15-11
15.2.3.2	Creating a Validation for an Entire Page	15-14
15.2.3.3	Creating a Validation for a Tabular Form Column	15-16
15.2.3.4	Creating a Validation for a Tabular Form Row	15-17
15.2.3.5	Editing a Validation in Component View	15-18
15.2.4	About Determining When Validations Execute	15-19
15.2.5	About Defining How Validation Error Messages Display	15-20

15.2.6	Processing Validations Conditionally	15-20
15.2.7	About Altering the Go to Error Link	15-21
15.2.8	What Happens When a Validation Fails?	15-21
15.3	Understanding Page Processes	15-22
15.3.1	About Page Processes	15-22
15.3.2	Managing Page Processes in Page Designer	15-22
15.3.2.1	Creating Page Processes in Page Designer	15-23
15.3.2.2	Editing Page Process Attributes in Page Designer	15-24
15.3.2.3	Controlling When a Page Process Executes in Page Designer	15-25
15.3.2.4	Creating Page Process Success and Error Messages in Page Designer	15-25
15.3.2.5	Making a Page Process Conditional in Page Designer	15-26
15.3.3	Managing Page Processes in Legacy Component View	15-26
15.3.3.1	Creating Page Processes in Component View	15-27
15.3.3.2	Editing Page Process Attributes in Component View	15-29
15.3.3.3	Controlling When a Page Process Executes	15-30
15.3.3.4	Creating Page Process Success and Error Messages	15-30
15.3.3.5	Making a Page Process Conditional	15-31
15.4	About Branches	15-32
15.5	Understanding Application Processes	15-32
15.5.1	On Demand Application Processes	15-32
15.5.2	About Application Processes that Execute On New Instance	15-33
15.5.3	About Running an On Demand Process from a Page Request	15-33
15.5.4	Application Process Example	15-34
15.5.5	Creating an Application Process	15-34
15.5.6	Editing Application Process Attributes	15-35
15.5.7	Creating Application Process Error Messages	15-36
15.5.8	Viewing the Application Processes History Report	15-36
15.6	Understanding Application Computations	15-37
15.6.1	About Application Computations	15-37
15.6.2	About Application Computations that Execute On New Instance	15-38
15.6.3	Creating an Application Computation	15-38
15.6.4	Accessing the Application Computation History Report	15-39
15.7	Using the Attribute Dictionary	15-40
15.7.1	About the Attribute Dictionary	15-40
15.7.2	Accessing the Manage Attribute Dictionary	15-40

16 Managing Shared Components

16.1	Working with Shared Components	16-1
16.1.1	About the Advantages of Using Shared Components	16-2
16.1.2	Accessing the Shared Components Page	16-2

16.1.3	About the Shared Components Icon	16-2
16.1.4	Shared Components Page	16-3
16.1.4.1	Application Logic	16-3
16.1.4.2	Security	16-4
16.1.4.3	Other Components	16-5
16.1.4.4	Navigation	16-5
16.1.4.5	User Interface	16-6
16.1.4.6	Files	16-7
16.1.4.7	Data References	16-7
16.1.4.8	Reports	16-7
16.1.4.9	Globalization	16-8
16.1.4.10	Tasks Region	16-8
16.2	Managing Component Settings	16-9
16.2.1	Configuring Component Settings	16-9
16.2.2	Integrating with Google Calendar	16-10
16.2.3	Configuring Color Picker Display	16-10
16.2.4	Configuring Interactive Report Action Menu Structure	16-11
16.2.5	Dynamically Adding Sliders to Regions	16-11
16.2.6	Configuring Switch Defaults	16-11
16.2.7	Viewing the Component Settings History Report	16-12
16.3	Managing Static Application Files	16-12
16.3.1	Specifying the Location of Static Application Files	16-13
16.3.2	Uploading Static Application Files	16-13
16.3.3	Viewing the Static Application File Report	16-14
16.3.4	Referencing Static Application Files	16-14
16.3.5	Downloading Static Application Files	16-15
16.3.5.1	Downloading a Static Application File	16-15
16.3.5.2	Downloading All Static Application Files in a Zip	16-15
16.3.6	Deleting Static Application Files	16-16
16.3.6.1	Deleting a Static Application File	16-16
16.3.6.2	Deleting All Static Application Files	16-16
16.4	Managing Static Workspace Files	16-17
16.4.1	Uploading Static Workspace Files	16-17
16.4.2	Viewing Static Workspace File Report	16-18
16.4.3	About Referencing Static Workspace Files	16-18
16.4.4	Downloading Static Workspace Files	16-18
16.4.4.1	Downloading a Static Workspace File	16-19
16.4.4.2	Downloading All Static Workspace Files in a Zip	16-19
16.4.5	Deleting Static Workspace Files	16-19
16.4.5.1	Deleting a Static Workspace File	16-19
16.4.5.2	Deleting All Static Workspace Files	16-20

16.5	Managing Application-Level Items	16-20
16.5.1	Creating an Application-Level Item	16-21
16.5.2	Viewing Application Item Usage	16-22
16.5.3	Editing Application-Level Item Attributes	16-23
16.5.4	About Creating an Application-Level Item that Functions as a Global Variable	16-23
16.6	Creating Lists of Values at the Application Level	16-24
16.6.1	What is a List of Values?	16-24
16.6.2	Creating a Named LOV at the Application Level	16-25
16.6.3	About Static LOVs	16-25
16.6.4	Editing an Existing LOV	16-25
16.6.4.1	Editing a LOV	16-25
16.6.4.2	Editing LOVs in Bulk	16-26
16.6.5	About Referencing Session State Within an LOV	16-26
16.6.6	Accessing LOV Reports	16-26
16.7	Creating Lists	16-27
16.7.1	About Creating Lists	16-27
16.7.2	About Navigation Menus	16-28
16.7.3	Creating Static Lists	16-29
16.7.3.1	About the Process of Creating a Static List	16-29
16.7.3.2	Creating a Static List from Scratch	16-30
16.7.3.3	About Adding Entries and Sublists to a Static List	16-31
16.7.3.4	Adding Entries or Sublists from Scratch	16-31
16.7.3.5	Copying Static List Entries Between Lists	16-33
16.7.3.6	Reparenting Static List Entries	16-34
16.7.3.7	Managing Orphaned Static List Entries	16-34
16.7.3.8	Resequencing Static List Entries	16-35
16.7.4	Creating Dynamic Lists	16-35
16.7.4.1	About the Process of Creating a Dynamic List	16-35
16.7.4.2	Understanding Dynamic List Syntax	16-36
16.7.4.3	Creating a Dynamic List from Scratch	16-37
16.7.5	Copying a List	16-38
16.7.5.1	Copying a List from the Current Application	16-38
16.7.5.2	Copying a List Between Applications	16-39
16.7.6	Adding a List to a Page	16-39
16.7.7	Editing List Attributes	16-40
16.7.8	Accessing List Reports	16-41
16.8	Using Shortcuts	16-42
16.8.1	What is a Shortcut?	16-42
16.8.2	Understanding Shortcut Types	16-42
16.8.2.1	Supported Shortcut Types	16-42

16.8.2.2	Text with JavaScript Escaped Single Quotes	16-43
16.8.2.3	Message	16-43
16.8.2.4	Message with JavaScript Escaped Single Quotes	16-43
16.8.3	Defining and Referencing Shortcuts	16-43
16.8.3.1	Defining a Shortcut from Scratch	16-43
16.8.3.2	Copying an Existing Shortcuts	16-44
16.8.3.3	Referencing a Shortcut	16-45
16.8.4	Editing Existing Shortcuts	16-45
16.8.5	Accessing Shortcut Reports	16-45
16.9	Managing Report Output	16-46
16.9.1	Printing a Report Region by Defining a Report Query	16-46
16.9.1.1	About Report Queries	16-46
16.9.1.2	Creating a Report Query	16-46
16.9.1.3	Editing a Report Query	16-49
16.9.1.4	Copying a Report Query	16-49
16.9.2	Formatting a Report Region or Report Query Using Report Layouts	16-49
16.9.2.1	About Report Layouts	16-50
16.9.2.2	About Report Layout Options	16-50
16.9.2.3	Creating a Report Layout	16-51
16.9.2.4	Editing a Report Layout	16-52
16.9.2.5	Copying a Report Layout	16-52

17 Managing Application Data

17.1	Creating Applications with Data Loading Capability	17-1
17.1.1	About Creating a Data Load Wizard	17-1
17.1.2	Creating a Data Load Wizard	17-3
17.1.3	Re-creating a Data Load Pages	17-5
17.1.4	Editing the Data Load Definition	17-5
17.2	About Using Collections	17-7
17.3	Implementing Web Services	17-8
17.3.1	About Web Services	17-8
17.3.2	Creating Web Service References in Oracle Application Express	17-9
17.3.2.1	Accessing the Web Service References Page	17-9
17.3.2.2	Specifying an Application Proxy Server Address	17-10
17.3.3	About Working with SSL Enabled Web Services	17-10
17.3.4	Creating Web Service References Based on a WSDL	17-10
17.3.4.1	About Creating Web Service References Based on a WSDL	17-11
17.3.4.2	Creating a Web Service Reference by Specifying a WSDL Document	17-11
17.3.4.3	Testing a Web Service Reference Created from a WSDL	17-12
17.3.5	Creating Web Service References Manually	17-12

17.3.5.1	About Creating Web Service References Manually	17-13
17.3.5.2	Creating a Web Service Reference Manually	17-13
17.3.5.3	Testing a Web Service Reference Created Manually	17-14
17.3.6	Creating RESTful Web Service References	17-14
17.3.6.1	About Creating RESTful Web Service References	17-14
17.3.6.2	Creating a RESTful Web Service Reference	17-15
17.3.6.3	Testing a REST Web Service Reference	17-16
17.3.7	Using the Web Service Reference Repository	17-17
17.3.8	Creating an Input Form and Report on a Web Service	17-17
17.3.8.1	About the Create Form and Report on Web Service Wizard	17-18
17.3.8.2	Creating a Form and Report After Creating a Reference	17-18
17.3.8.3	Creating a Form and Report by Adding a New Page	17-19
17.3.9	Creating a Form on a Web Service	17-20
17.3.9.1	About the Create Form on Web Service Wizard	17-20
17.3.9.2	Creating a Form After Creating a Reference	17-21
17.3.9.3	Creating a Form by Adding a New Page	17-22
17.3.10	How to Invoke a Web Service as a Process	17-23
17.3.10.1	Invoking a Web Service as a Process	17-24
17.3.10.2	Displaying Web Service Results in a Report	17-25
17.3.11	Editing a Web Service Process	17-26
17.3.12	Viewing a Web Service Reference History	17-27
17.3.13	Exposing a Report Region as a RESTful Web Service	17-27
17.3.13.1	About Exposing a Report Region as a RESTful Web Service	17-28
17.3.13.2	Enabling RESTful Access to a Report Region	17-28
17.3.13.3	About Accessing a RESTful Enabled Report Region from a Web Service Client	17-29
17.4	Accessing Data with Database Links	17-30
17.4.1	About Database Links	17-31
17.4.2	Creating a Database Link	17-31
17.4.3	Viewing an Existing Database Link	17-31
17.5	About DML Locking	17-32
17.5.1	About DML Locking	17-32
17.5.2	APEX_DML_LOCK_WAIT_TIME	17-32
17.5.3	FSP_DML_LOCK_ROW	17-32

18 Extending Application Capabilities

18.1	Sending Email from an Application	18-1
18.1.1	About Configuring Email	18-1
18.1.1.1	About Enabling Network Services	18-2
18.1.1.2	About Configuring Email in a Full Development Environment	18-2
18.1.2	About Email Provisioning	18-2

18.1.3	About the Mail Queue	18-3
18.1.4	About the APEX_MAIL Package	18-3
18.2	Implementing Plug-ins	18-3
18.2.1	About Plug-ins	18-4
18.2.2	Viewing Plug-in Examples	18-4
18.2.3	Accessing the Plug-ins Page	18-5
18.2.4	Creating a Plug-in	18-5
18.2.5	Editing a Plug-in	18-7
18.2.6	Adding Custom Attributes to a Plug-in	18-7
18.2.7	Uploading Files Associated with a Plug-in	18-8
18.2.7.1	Uploading a File to Associate with a Plug-in	18-8
18.2.7.2	Automatically Loading CSS and JavaScript Files	18-9
18.2.8	Adding Events to a Plug-in	18-10
18.2.9	Deleting a Plug-in	18-10
18.2.10	Viewing the Plug-in Repository	18-11
18.2.11	Importing a Plug-in from the Plug-in Page	18-11
18.2.12	Exporting a Plug-in from the Plug-in Page	18-12
18.2.13	Resetting the Plug-in Interactive Report	18-12
18.2.14	Viewing Plug-in Utilization Page	18-12
18.2.15	Viewing Plug-in History	18-13
18.3	Understanding jQuery and jQuery UI Support	18-13
18.3.1	About Compatibility Issues When Upgrading from Application Express Release 4.X to 5.1	18-13
18.3.2	About Available jQuery UI Features	18-15
18.3.2.1	About Accessing Other jQuery UI Components	18-16
18.3.3	Referencing the jQuery Library in Your JavaScript Code	18-16
18.3.3.1	About Managing JavaScript Code in Your Application	18-17
18.3.3.2	About Using JavaScript Code in a Plug-In	18-18
18.4	Manually Refreshing Oracle Application Express Components	18-19

19 Managing Application Security

19.1	Understanding Administrator Security Best Practices	19-1
19.1.1	About Oracle Application Express Administrator Roles	19-2
19.1.2	About Configuring Oracle REST Data Services with Oracle Application Express	19-2
19.1.3	About Configuring Oracle HTTP Server with mod_plsql with Oracle Application Express	19-3
19.1.4	About Security Considerations When Using the Embedded PL/SQL Gateway	19-4
19.1.5	Managing Instance Security	19-4
19.1.5.1	About Creating Login Controls	19-5

19.1.5.2	About Enabling Public File Upload	19-5
19.1.5.3	About Restricting User Access by IP Address	19-5
19.1.5.4	About Specifying an Instance Proxy	19-6
19.1.5.5	About Utilizing Secure Sockets Layer	19-6
19.1.5.6	About Enabling RESTful Access	19-7
19.1.5.7	About Rejoin Sessions	19-7
19.1.5.8	About Isolating Workspaces	19-8
19.1.5.9	About Utilizing Session Timeout	19-8
19.1.5.10	Restricting Password Reuse	19-9
19.1.5.11	About Using SQL in Websheets	19-10
19.1.5.12	About Enabling RESTful Services	19-10
19.1.6	Configuring Instance Settings	19-10
19.1.6.1	About Enabling Self Service Workspace Provisioning	19-11
19.1.6.2	About Enabling and Configuring Email	19-11
19.1.6.3	About Configuring Storage Options	19-12
19.1.6.4	About Creating Wallets	19-12
19.1.7	About Configuring Workspace Purge Settings	19-12
19.1.8	Understanding Workspace Management	19-12
19.1.8.1	About Managing Workspace to Schema Assignments	19-13
19.1.8.2	About Managing Developers and Users	19-13
19.1.8.3	About Managing Component Availability	19-13
19.1.9	About Integrating with Oracle BI Publisher	19-14
19.1.10	About the Advantages of the Application Express Runtime Environment	19-14
19.1.11	Enabling Network Services in Oracle Database 11g or Later	19-14
19.1.11.1	When and Why Network Services Must be Enabled	19-15
19.1.11.2	Granting Connect Privileges Prior to Oracle Database 12c	19-15
19.1.11.3	Granting Connect Privileges in Oracle Database 12c	19-17
19.1.11.4	Troubleshooting an Invalid ACL Error	19-17
19.2	Understanding Developer Security Best Practices	19-18
19.2.1	About Items of Type Password	19-19
19.2.2	Identifying At Risk Password Items	19-19
19.2.2.1	Viewing the Security Profiles Report	19-20
19.2.2.2	Viewing the Password Items Report	19-20
19.2.3	Understanding Cross-Site Scripting Protection	19-21
19.2.3.1	What Is Cross Site-scripting Security Breach?	19-21
19.2.3.2	Protecting HTML Regions and Other Static Areas	19-21
19.2.3.3	About Protecting Dynamic Output	19-23
19.2.3.4	About Protecting Report Regions	19-24
19.2.3.5	About Protecting Form Items	19-25
19.2.3.6	About Restricting Characters Entered on an Item	19-25

19.2.4	About Session State and Security	19-26
19.2.5	Understanding Session State Protection	19-26
19.2.5.1	How Session State Protection Works	19-27
19.2.5.2	Enabling Session State Protection	19-27
19.2.5.3	Configuring Session State Protection	19-29
19.2.6	About Securing File Uploads	19-39
19.3	Controlling Access to Applications, Pages, and Page Components	19-39
19.3.1	How the Access Control List Works	19-40
19.3.2	Creating an Access Control List	19-40
19.3.2.1	Creating an Access Control from the Application Home Page	19-40
19.3.2.2	Creating an Access Control While Editing a Page	19-41
19.3.3	Selecting an Application Mode and Adding Users	19-42
19.3.3.1	Selecting an Application Mode	19-42
19.3.3.2	Adding Users to an Access Control List and Selecting Privileges	19-43
19.3.3.3	Removing Users from the Access Control List	19-44
19.3.4	About Controlling Access for Pages and Page Components	19-44
19.4	Establishing User Identity Through Authentication	19-45
19.4.1	Understanding Authentication	19-45
19.4.1.1	How Authentication Works	19-46
19.4.1.2	About Support for Deep Linking	19-46
19.4.1.3	About Determining Whether to Include Authentication	19-47
19.4.2	Creating an Authentication Scheme	19-47
19.4.3	Understanding Preconfigured Authentication Schemes	19-48
19.4.3.1	Application Express Accounts	19-48
19.4.3.2	Custom Authentication	19-49
19.4.3.3	Database Accounts	19-51
19.4.3.4	HTTP Header Variable	19-52
19.4.3.5	LDAP Directory Verification	19-53
19.4.3.6	No Authentication (using DAD)	19-54
19.4.3.7	Open Door Credentials	19-55
19.4.3.8	Oracle Application Server Single Sign-On Server	19-56
19.4.4	Managing Existing Authentication Schemes	19-57
19.4.5	Viewing the Authentication Scheme Associated with an Application	19-58
19.4.6	Changing the Authentication Scheme Associated with an Application	19-58
19.4.7	Creating a Login Page	19-59
19.4.8	Viewing Authentication Scheme Reports	19-59
19.5	Providing Security Through Authorization	19-59
19.5.1	How Authorization Schemes Work	19-60
19.5.2	Creating and Editing an Authorization Scheme	19-60
19.5.2.1	About Authorization Scheme Types	19-60
19.5.2.2	Creating an Authorization Scheme	19-61

19.5.2.3	Editing Attributes of an Existing Authorization Scheme	19-62
19.5.2.4	Changing the Evaluation Point Attribute	19-62
19.5.2.5	About Resetting Authorization Scheme State	19-63
19.5.3	Attaching an Authorization Scheme to an Application, Page, or Components	19-63
19.5.3.1	Attaching an Authorization Scheme to an Application	19-64
19.5.3.2	Attaching an Authorization Scheme to a Page	19-64
19.5.3.3	Attaching an Authorization Scheme to a Control or Component	19-64
19.5.4	Viewing Authorization Reports	19-65

20 Managing Application Globalization

20.1	Understanding Application Translation and Globalization Support	20-1
20.1.1	About the Translation Process	20-1
20.1.2	About Language Identification	20-2
20.1.3	About the Rules for Translating Database Applications	20-2
20.1.4	How Translated Applications Are Rendered	20-2
20.1.5	Making Application Attributes Translatable	20-3
20.1.5.1	About Shortcuts that Support Translatable Messages	20-3
20.1.5.2	About Messages	20-3
20.1.5.3	About Dynamic Translation Text Strings	20-4
20.1.5.4	About Translating Region Titles	20-4
20.1.5.5	About Translating Templates	20-5
20.2	Specifying the Primary Language for an Application	20-5
20.3	Using Format Masks for Items	20-6
20.3.1	About Apply Format Masks to Items	20-7
20.3.2	Example of Editing Item Attributes	20-7
20.4	About Translating Applications for Multibyte Languages	20-7
20.5	Understanding the Translation Process	20-7
20.5.1	Step 1: Map the Target Language	20-8
20.5.2	Step 2: Seed and Download to a Translation File	20-9
20.5.2.1	Seeding Translatable Text	20-9
20.5.2.2	Downloading an XLIFF File	20-9
20.5.3	Step 3: Translate the XLIFF File	20-11
20.5.4	Step 4: Upload and Apply a Translated XLIFF Document	20-12
20.5.4.1	Uploading a Translated XLIFF Document	20-12
20.5.4.2	Applying an Uploaded XLIFF Document	20-12
20.5.4.3	Deleting an Uploaded XLIFF Document	20-13
20.5.5	Step 5: Publish the Application	20-14
20.5.6	Manually Editing a Translation	20-14
20.6	Translating Messages	20-15
20.6.1	About Translating Messages	20-16

20.6.2	Translating Messages Used in PL/SQL Procedures	20-16
20.6.2.1	About Translating Messages Used in PL/SQL Procedures	20-16
20.6.2.2	Defining a Translation Message	20-16
20.6.3	Interactive Grid Messages Requiring Translation	20-17
20.6.4	Translating Messages Used Internally by Oracle Application Express	20-35
20.6.4.1	Creating Text Messages	20-36
20.6.4.2	Editing Text Messages	20-36
20.6.4.3	Internal Messages Requiring Translation	20-37
20.6.4.4	Interactive Report Messages Requiring Translation	20-41
20.7	Translating Data That Supports List of Values	20-58
20.7.1	Defining a Dynamic Translation	20-58
20.8	Understanding Supported Globalization Codes	20-59
20.8.1	Oracle Application Express Globalization Codes	20-59

21 Managing Application Performance

21.1	About Best Practices to Improve Performance	21-1
21.2	Identifying Performance Issues	21-2
21.2.1	About Troubleshooting Slow Running Applications	21-2
21.2.2	Diagnosing Performance Issues	21-2
21.2.3	About Debugging Problematic SQL Queries	21-3
21.2.4	About Addressing Slow Queries	21-3
21.3	Utilizing Logs and Reports	21-3
21.3.1	How to Manage Log Files	21-4
21.3.2	Viewing the Application Dashboard Report	21-4
21.3.3	Workspace Activity Reports	21-5
21.3.4	Instance Activity Reports	21-5
21.4	About Utilizing Database Reporting	21-6
21.5	About Database Parameters that Impact Performance	21-6
21.6	About Limiting Resources	21-7
21.7	About Uploading Static Files to Your Web Server	21-7
21.8	Creating Custom Activity Reports Using APEX_ACTIVITY_LOG	21-8
21.8.1	Enabling Logging for an Application	21-8
21.8.2	APEX_ACTIVITY_LOG	21-9
21.8.3	APEX_ACTIVITY_LOG Sample Query	21-10

22 Debugging an Application

22.1	Running Advisor to Check Application Integrity	22-1
22.1.1	About Oracle Application Express Advisor	22-1
22.1.2	Running Advisor on an Entire Application	22-1

22.1.3	Running Advisor on a Single Page	22-2
22.2	Reviewing Session State	22-3
22.3	About Monitoring Application and Page Resource Use	22-3
22.4	Enabling SQL Tracing and Using TKPROF	22-3
22.5	Utilizing Debug Mode	22-4
22.5.1	About Debug Mode	22-4
22.5.2	Enabling and Disabling Debug Mode for an Application	22-5
22.5.3	Running an Application in Debug Mode	22-5
22.5.4	Viewing Debug Reports in Development Mode	22-6
22.5.5	About Using f?p Syntax to Access Debug Mode	22-6
22.5.6	Viewing Debug Reports from a Running Application	22-7
22.5.7	Viewing Debug Reports	22-7
22.6	About Removing Controls and Components to Isolate a Problem	22-8

23 Deploying an Application

23.1	About Suggested Environments	23-2
23.2	System Development Life Cycle Methodologies to Consider	23-2
23.2.1	About Iterative Vs Planned Development	23-2
23.2.2	About the Advantages of Creating Prototypes	23-3
23.2.3	Waterfall	23-3
23.2.4	Spiral	23-4
23.2.5	Rapid Application Development	23-4
23.3	Understanding the Deployment Process	23-4
23.3.1	About Moving an Application to Another Instance	23-4
23.3.2	Deployment Options to Consider	23-5
23.3.3	About Whether to Copy a Workspace	23-5
23.3.4	About Whether to Copy a Database	23-5
23.3.5	About the Application ID	23-6
23.4	How to Move an Application to Another Instance	23-6
23.4.1	About Managing Database Objects	23-7
23.5	How to Create a Custom Packaged Application	23-8
23.5.1	How Creating a Custom Packaged Application Simplifies Deployment	23-8
23.5.2	Creating a Custom Packaged Application	23-9
23.5.2.1	Accessing the Supporting Objects Page	23-9
23.5.2.2	Supporting Objects Page	23-9
23.5.3	Adding an Access Control List to a Custom Packaged Application	23-12
23.5.4	Installing Supporting Objects	23-13
23.5.5	Deleting Supporting Objects Scripts, Messages, and Installation Options	23-13
23.5.6	Upgrading a Custom Packaged Application	23-14
23.5.6.1	Defining an Upgrade Script	23-14

23.5.6.2	Upgrading a Custom Packaged Application	23-15
23.5.7	Deinstalling Supporting Objects	23-15
23.5.8	Viewing an Install Summary	23-16
23.6	Exporting an Application and Application Components	23-16
23.6.1	About the Export Process	23-17
23.6.2	About The Export Page	23-17
23.6.3	Exporting an Application	23-19
23.6.4	Exporting Application Components	23-21
23.6.5	Exporting a Workspace	23-23
23.6.6	Exporting a Page in an Application	23-24
23.6.7	Exporting a Websheet	23-25
23.6.8	About Exporting Static Files	23-25
23.6.9	Exporting Themes	23-26
23.6.10	Exporting Plug-ins	23-26
23.6.11	Exporting User Interface Defaults	23-27
23.6.11.1	About Exporting User Interface Defaults	23-27
23.6.11.2	Exporting User Interface Defaults from the Export Page	23-28
23.6.11.3	Exporting User Interface Defaults from the User Interface Defaults Page	23-28
23.6.12	Exporting Team Development Feedback	23-29
23.6.13	Exporting Script Files from the Script Repository	23-30
23.7	Importing Export Files	23-30
23.7.1	Importing an Application, Page or Component Export	23-31
23.7.2	About Importing Application Groups	23-32
23.7.3	Importing a Websheet	23-32
23.7.4	Importing Plug-ins	23-33
23.7.5	Importing Themes	23-33
23.7.6	Importing User Interface Defaults	23-34
23.7.7	Importing Team Development Feedback	23-35
23.8	Installing Export Files	23-35
23.8.1	About Installing Export Files	23-35
23.8.2	Accessing the Export Repository	23-36
23.8.3	Installing an Application Export from the Export Repository	23-36
23.8.4	Installing Other Files from the Export Repository	23-37
23.8.5	Deleting Files from the Export Repository	23-38
23.8.6	Installing Export Files from SQL*Plus	23-38
23.8.7	Using SQL*Plus to Install Export Files	23-38
23.9	About Publishing the Database Application URL	23-38
23.10	About Publishing the Websheet Application URL	23-39
23.11	Using Build Options to Control Configuration	23-40
23.11.1	Creating Build Options	23-41

23.11.2	Managing Build Options	23-41
23.11.3	Selecting a Build Option	23-41
23.11.4	Exporting Build Options or Build Option Status	23-42
23.11.5	Viewing the Build Option Utilization Report	23-43

24 Managing Application Legacy Components

24.1	Managing Pages in Legacy Component View	24-1
24.1.1	Legacy Component View Window	24-2
24.1.2	Switching from Page Designer to Legacy Component View	24-3
24.1.3	Understanding Legacy Component View UI Elements	24-3
24.1.3.1	Component View Toolbar	24-4
24.1.3.2	Component View Navigation Bar	24-5
24.1.3.3	Page Rendering	24-7
24.1.3.4	Page Processing	24-8
24.1.3.5	Shared Components	24-8
24.1.4	Using Legacy Component View	24-10
24.1.4.1	Viewing a Page in Legacy Component View	24-10
24.1.4.2	Edit All, Copy, and Create Icons	24-11
24.1.4.3	Creating a Page Control in Legacy Component View	24-12
24.1.5	Accessing Page Attributes in Legacy Component View	24-12
24.1.5.1	Editing Page Attributes in Component View	24-13
24.1.5.2	Page Attributes in Component View	24-13
24.1.5.3	Incorporating JavaScript Using Component View	24-25
24.1.5.4	Configuring Rejoin Sessions in Component View	24-26
24.1.5.5	Configuring Dialog Attributes in Component View	24-27
24.1.6	Running a Page in Legacy Component View	24-28
24.1.7	Runtime Developer Toolbar	24-29
24.1.8	Copying a Database Application Page in Legacy Component View	24-30
24.1.9	Deleting Pages in Legacy Component View	24-31
24.1.9.1	Deleting a Page in Component View	24-31
24.1.9.2	Deleting a Page While Editing Page Attributes	24-31
24.1.9.3	Deleting Multiple Pages	24-31
24.1.9.4	Deleting a Range of Pages	24-32
24.2	Managing Legacy Charts	24-32
24.2.1	About Chart Types	24-32
24.2.1.1	About HTML5 Charts	24-33
24.2.1.2	About Flash Charts	24-33
24.2.2	Creating SQL Queries for Charts	24-33
24.2.2.1	Standard Chart Syntax	24-33
24.2.2.2	Dial Chart Syntax (Flash and HTML5)	24-34

24.2.2.3	Multiple Series Syntax (Flash and HTML5)	24-34
24.2.2.4	Range Chart Syntax (Flash and HTML5)	24-34
24.2.2.5	Scatter Chart Syntax (Flash and HTML5)	24-34
24.2.2.6	Candlestick Chart Syntax (Flash and HTML5)	24-35
24.2.2.7	Gantt Chart Syntax (Flash Only)	24-35
24.2.3	Adding a Legacy Chart to a New Page	24-35
24.2.4	Managing Charts in Page Designer	24-37
24.2.4.1	Adding a Chart in Page Designer	24-37
24.2.4.2	Editing Chart Attributes in Page Designer	24-38
24.2.4.3	Switching an HTML5 Chart to a Flash Chart	24-39
24.2.4.4	Enabling Asynchronous Updates	24-39
24.2.5	Managing Charts in Legacy Component View	24-41
24.2.5.1	Adding a Chart in Component View	24-41
24.2.5.2	Switching an HTML5 Chart to a Flash Chart	24-43
24.2.5.3	Editing Chart Attributes in Component View	24-43
24.2.5.4	Enabling Asynchronous Updates	24-44
24.2.6	Understanding SVG Chart Support	24-45
24.2.6.1	Upgrading SVG Charts to HTML5	24-45
24.2.6.2	Understanding SVG Chart Cascading Style Sheet Classes	24-46
24.2.6.3	Referencing a Custom SVG Chart Cascading Style Sheet	24-49
24.2.6.4	Specifying Custom CSS Styles Inline for SVG Charts	24-50
24.2.6.5	Displaying SVG Charts in Other Languages	24-50
24.2.7	Understanding HTML Chart Support	24-51
24.2.7.1	Upgrading an HTML Chart to HTML5	24-51
24.2.7.2	Upgrading All HTML Charts to HTML5	24-52
24.2.8	About Saving Flash Charts	24-52
24.2.9	Using Custom XML with Flash and HTML5 Charts	24-52
24.3	Creating a Legacy Master Detail Form	24-53
24.4	Managing Legacy Tabular Forms	24-56
24.4.1	About Creating Tabular Forms	24-56
24.4.2	About Running a Tabular Form	24-57
24.4.3	How Tabular Forms Work	24-58
24.4.4	Creating a Legacy Tabular Form	24-59
24.4.5	Synchronizing Tabular Form Column Order with Region Source	24-61
24.5	Managing Legacy Calendars	24-61
24.5.1	Creating a Legacy Classic Calendar	24-62
24.5.2	Managing Calendars in Legacy Component View	24-64
24.5.2.1	Editing Calendar Attributes in Component View	24-65
24.5.2.2	Editing an Existing Calendar to Include Add and Edit Functionality	24-66
24.5.2.3	Enabling the Dragging and Dropping of Data In an Existing Legacy Calendar	24-67

24.5.3	About Supported Calendar Substitution Strings in Legacy Calendars	24-67
24.6	Viewing Legacy Components in Gallery	24-68

A About Item Types

A.1	Available Item Types	A-1
-----	----------------------	-----

B Oracle Application Express Limits

B.1	Component Limits	B-1
-----	------------------	-----

C Available Conditions

C.1	Conditions	C-1
C.2	Conditions Available in App Builder	C-1

D About Granted Privileges

D.1	About Granting Privileges to Database Users	D-1
D.2	Understanding Privileges Granted to PUBLIC	D-1
D.2.1	Packages	D-2
D.2.2	Procedures	D-4
D.2.3	Functions	D-4
D.2.4	Tables	D-4
D.2.5	Views	D-5
D.2.6	Sequences	D-9
D.2.7	Types	D-9

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.

Topics:

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

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:

Topics:

- [Changes in Oracle Application Express Release 5.1](#) (page xlv)

Changes in Oracle Application Express Release 5.1

The following are changes in *Oracle Application Express App Builder User's Guide* for Oracle Application Express release 5.1.

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

New Features

The following features are new in this release:

- Interactive Grid

Interactive grid presents end users with a set of data in a searchable, customizable report. From a functional perspective, an interactive grid includes the customizations capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.

Key features include:

- Full Featured Grid - Includes powerful reporting capabilities, including fixed headers, frozen columns, scroll pagination, multiple filters, sorting, aggregates, and computations. Includes powerful reporting capabilities, including fixed headers, frozen columns, scroll pagination, multiple filters, sorting, aggregates, and computations.
- Extensible and Customizable - Edit text, numerical data, and list of values. Interactive grids support all item types and item type plug-ins.
- Master Detail - Create master-detail relationships that are any number of levels deep and across. Create master-detail relationships that are any number of levels deep and across.

See "[Creating a Report](#) (page 8-5)," and "[Managing Interactive Grids](#) (page 8-13)," and "[Managing Master Detail Forms](#) (page 9-14)."

- Oracle JET Charts

Charts in Oracle Application Express release 5.1 are powered by the Oracle JET charting library. Oracle JET charts are a component of the Oracle JavaScript

Extension Toolkit (JET), an open source toolkit based on modern JavaScript, CSS3 and HTML5 design and development principles. Oracle JET charts are fast, customizable, and versatile. Charts are fully HTML5 capable and work on any modern browser, regardless of platform, screen size, or features. Easily migrate existing AnyChart charts using the Upgrade Application Wizard.

See "[Creating Charts](#) (page 10-23)."

- Wizard Simplification

In release 5.1, wizards have been streamlined and simplified.

- Create Application Wizard - The Create Application Wizard for Desktop applications now supports the creation of interactive grid pages as reports, forms, and master detail. When creating an application from a spreadsheet, this wizard also supports interactive grids as either a single page or a report and form page.

See "[Using the Create Application Wizard](#) (page 6-1)."

- Create Page Wizard - The Create Page Wizard features a more consistent, streamlined interface consisting of fewer steps. Master Detail forms now incorporate an interactive grid region on either a single page or on two pages.

"See [Managing Pages in a Database Application](#) (page 6-44)."

- Calendar

Calendar in this release offers many improvements:

- End Date Displayed Inclusive - In release 5.0, the CSS calendar considered the end date of an all-day event as *exclusive* (similar to the jQuery FullCalendar Plugin). In release 5.1, the end date is *inclusive* like all other Oracle Application Express components.
- JavaScript Customization - Developers can add JavaScript code to support customization of the FullCalendar initialization using the new Initialization JavaScript Code attribute.
- Dynamic Actions Events - Enables developers to capture events within the calendar and define dynamic actions against these events.
- Keyboard Support - When the calendar grid has focus, the arrow keys can be used to navigate within the calendar.

See "[Creating Calendars](#) (page 10-1)."

- Item Types

File Browse page items can be configured to support multiple file uploads, and can be restricted by file types.

- Page Designer

- 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. However, when you click a component and the Property Editor highlights the corresponding attribute in the right pane.
- Two Pane Mode - Page Designer now enables developers to view just two panes instead of three. Two Pane mode enables developers to focus on two panes at a time.

- Drag and Drop Tab Reordering - Developers can now customize Page Designer by reordering tabs across panes.
- Property Editor, Filter Properties Search - Search for a group or an attribute in the Property Editor by entering keyword in the Filter Properties field.
- Property Editor Change Indicator - Changed attributes display with a blue marker until the page is saved.

See "[Managing Pages in Page Designer](#) (page 7-1)."

- Page Submit
 - Reload On Submit Page Attribute - Enables developers to specify when the page should be reloaded following a page submission.
- Universal Theme and User Experience Enhancements
 - Font APEX - Font APEX is a new icon library designed specifically for Oracle Application Express and the Universal Theme. Font APEX includes over 1,000 icons. To learn more about Font APEX, install the *Universal Theme Sample Application*.
 - Live Template Options - Live Template Options enable developers to alter a component's template options without refreshing the page.
 - Theme Styles - Enabling users to select a theme style in a running application.

See "[Using Quick Edit to Modify Live Template Options](#) (page 11-40)" and "[Enabling Users to Select a Theme Style](#) (page 11-5)."

Deprecated and Desupported Features

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

Other Changes

The following are additional changes in the release:

- All content has been updated to reflect new functionality.
- Screen captures and graphics have been added and updated to reflect Oracle Application Express release 5.1 user interface enhancements.
- Changes to the *Oracle Application Express App Builder User's Guide*
 - A new organization in which legacy application components are grouped together. Legacy application components are components Oracle still supports but does not recommend using.
 - See "[Managing Application Legacy Components](#) (page 24-1)."
 - "Appendix E: Accessibility in Oracle Application Express" has been removed. Accessibility information can now be found in *Oracle Application Express Accessibility Guide*.

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*

Topics:

- [About Accessing Your Development Environment](#) (page 1-1)
- [Understanding Oracle Application Express](#) (page 1-2)
- [Understanding the Workspace Home Page](#) (page 1-3)
- [Understanding App Builder](#) (page 1-11)
- [Editing Your Account Profile](#) (page 1-22)
- [Editing User Preferences](#) (page 1-23)
- [Using Oracle Application Express Documentation](#) (page 1-25)

1.1 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 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"](#)

1.2 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?](#) (page 1-2)
- [How Oracle Application Express Works](#) (page 1-2)
- [About the Differences Between Runtime and Full Development Environments](#) (page 1-3)

1.2.1 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.

1.2.2 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

1.2.3 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. This is the appropriate choice for production implementations.
- **Full development environment.** In a full development environment, users can develop, modify, run, and delete Application Express applications. This environment 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*

1.3 Understanding the Workspace Home Page

When you sign in to Oracle Application Express development environment, the Workspace home page appears. 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 Workspace home page.

- [Workspace Home Page](#) (page 1-4)

- [Top Apps and Top Users Dashboards](#) (page 1-5)
- [News and Messages](#) (page 1-5)
- [Developer Navigation Tools](#) (page 1-6)
- [Available Updates Region](#) (page 1-11)
- [Accessibility Mode Region](#) (page 1-11)

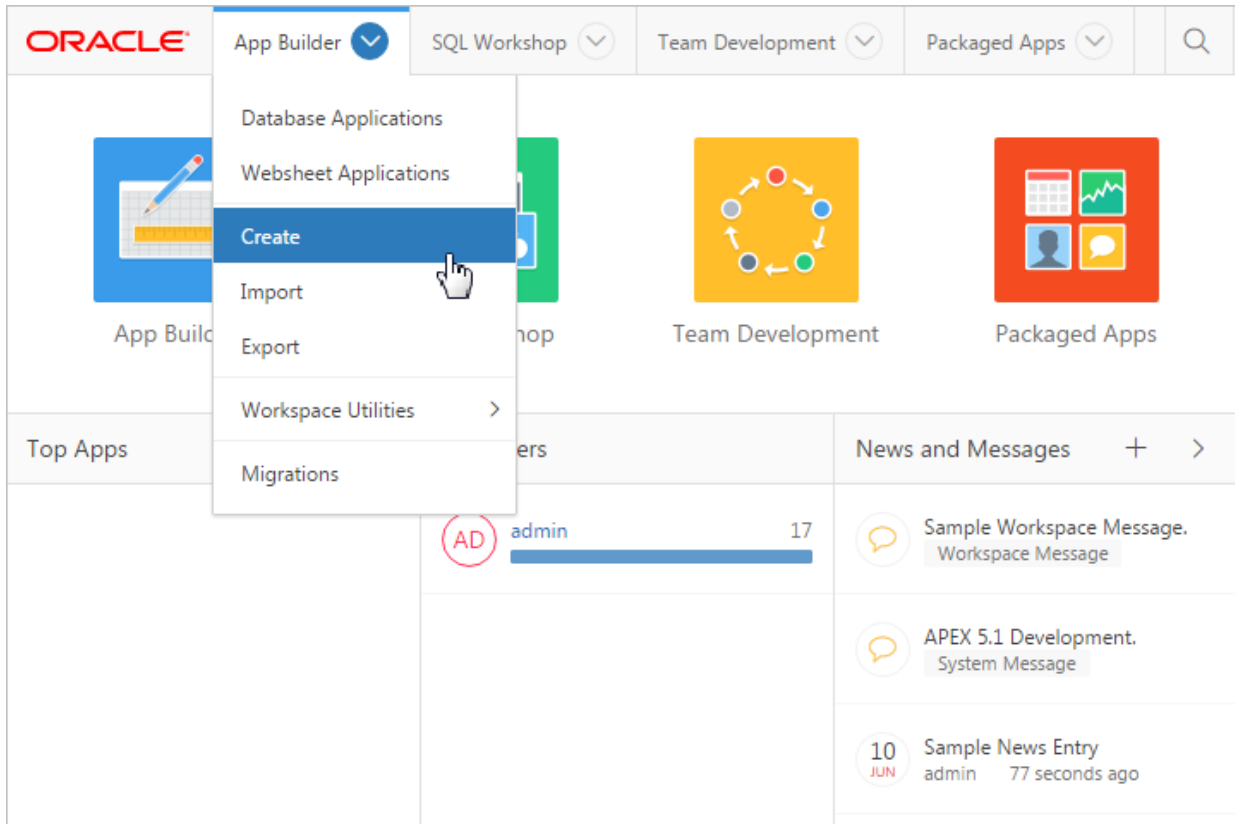
1.3.1 Workspace Home Page

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.
- **Packaged Apps** - Links to the Packaged Apps page. Install and run packaged applications to learn more about the types of applications you can build and how to construct specific types of pages.

ORACLE				App Builder	SQL Workshop	Team Development	Packaged Apps	Q
				App Builder	SQL Workshop	Team Development	Packaged Apps	
Top Apps	Top Users		News and Messages + >					
		admin	17			Sample Workspace Message. Workspace Message		
						APEX 5.1 Development. System Message		
						10 JUN	Sample News Entry admin 77 seconds ago	

An 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 Packaged Apps. The illustration that follows shows the **App Builder** menu.



 **See Also:**

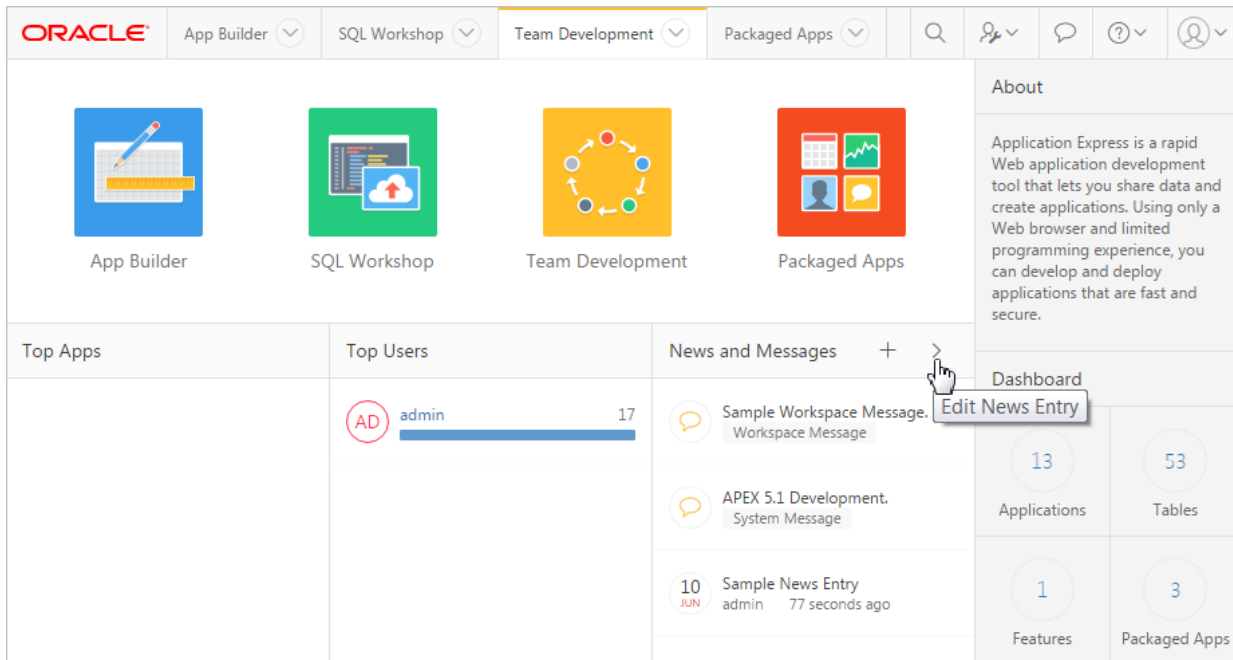
- "[App Builder Concepts](#) (page 2-1)"
- *Oracle Application Express SQL Workshop Guide*
- "[Managing the Application Life Cycle with Team Development](#) (page 4-1)"
- "[Utilizing Packaged Applications](#) (page 3-1)"

1.3.2 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.

1.3.3 News and Messages

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



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 \(page 4-28\)"](#)
- ["Defining a System Message" and "Defining a Workspace Announcement" in Oracle Application Express Administration Guide](#)

1.3.4 Developer Navigation Tools

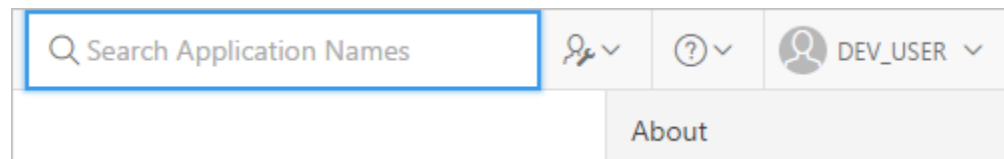
Developer Navigation tools display in the header region on many Oracle Application Express pages, including the Workspace home page, App Builder home page, the Application home page, Page Designer, Component View, and Shared Components.

- [Global Search Field \(page 1-7\)](#)

- [Administration Menu](#) (page 1-8)
- [Account Menu](#) (page 1-10)

1.3.4.1 Global Search Field

The Search field displays in the header region in upper right corner of most App Builder pages.



How the search field works depends upon the page you are on. Consider the following examples:

- Workspace home page - Search for application names.
- App Builder home page - Search for application metadata across all applications.
- Application home page - Search for application metadata across the current application.
- Team Development home page - Search the Team Development home page and other pages within Team Development.
- Workspace Administration home page - Search for users.

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

```
page  
application_ID:page  
application_ID-page  
regexp:expression
```

Where:

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

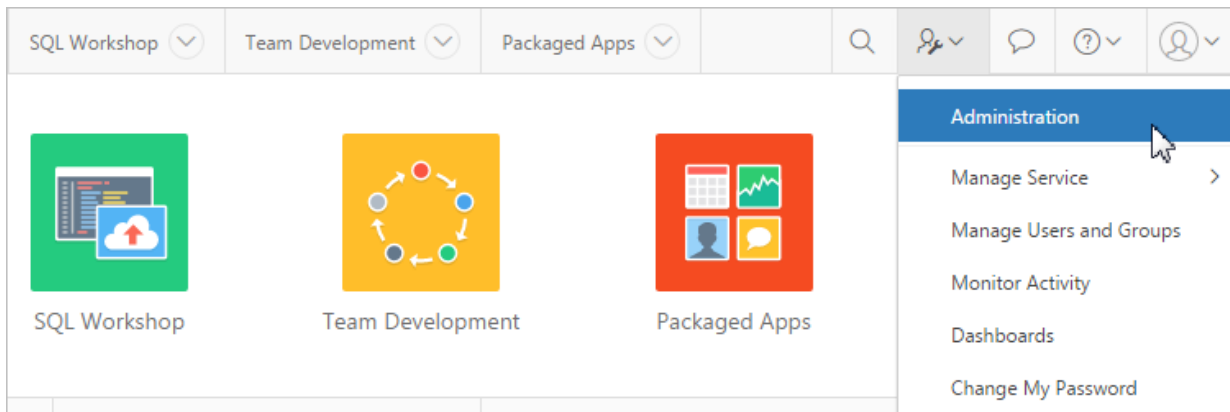
Where:

- *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


1.3.4.2 Administration Menu



The Administration menu displays to the right of the Global Search field and features an outline person with a wrench. Clicking this 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, Websheet application development, application development, and view database details.

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

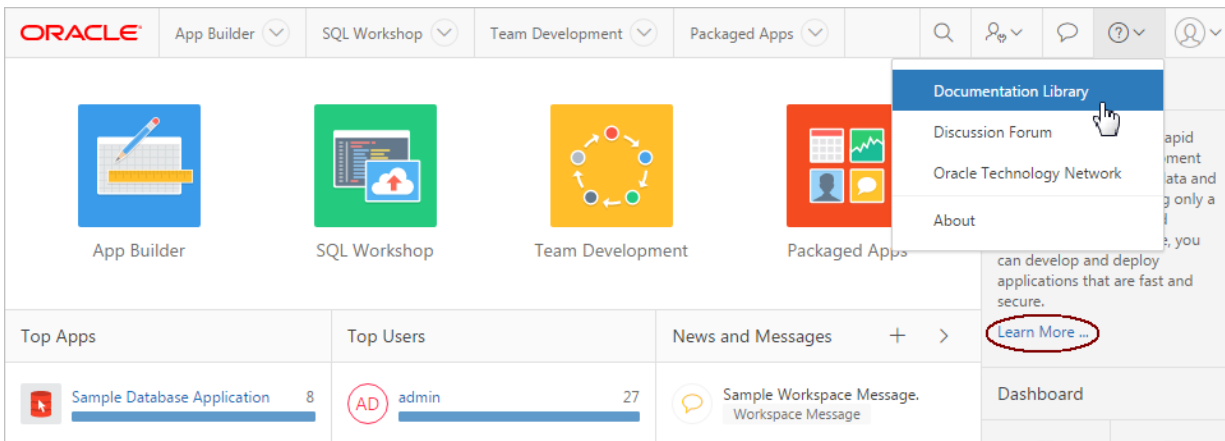
 **See Also:**

- "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 Password \(page 1-23\)"](#)

- [Help Menu \(page 1-9\)](#)

1.3.4.2.1 Help Menu

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



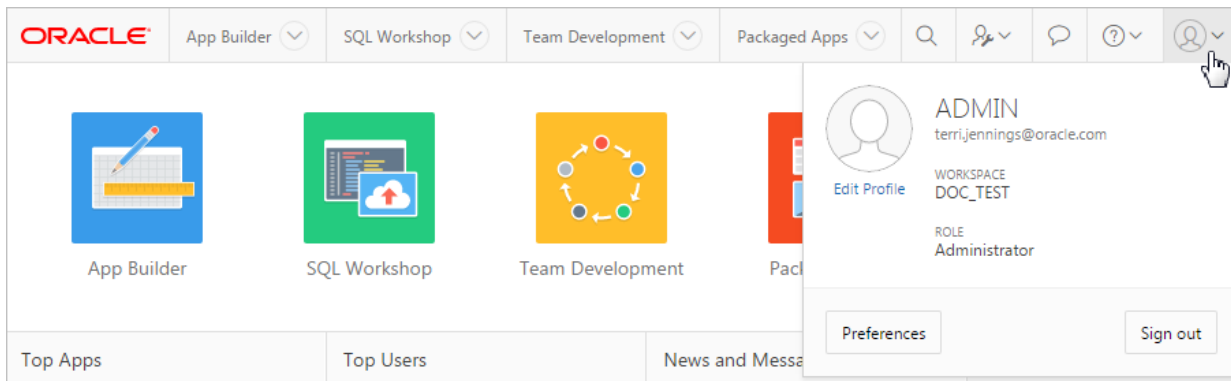
Click the Help menu to access the following menu items:

- **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 \(page 1-25\)"](#)

1.3.4.3 Account Menu



The Account menu displays in the upper right corner of the page and features an outline of a person. The Account menu displays 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:**

- ["Editing Your Account Profile \(page 1-22\)"](#)
- ["Editing User Preferences \(page 1-23\)"](#)

1.3.5 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>

1.3.6 Accessibility Mode Region

The Accessibility Mode list displays in the lower right side of the page. Use the Accessibility Mode list to identify if a page is optimized for screen readers. Available options include: Standard, Screen Reader, and High Contrast.

See Also:

Oracle Application Express Accessibility Guide

1.4 Understanding App Builder

Developers use App Builder to create the pages that make up an Oracle Application Express application. 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](#) (page 1-12)
- [Using the App Builder Home Page](#) (page 1-13)
- [Accessing the Application Home Page](#) (page 1-16)
- [Using the Application Home Page](#) (page 1-17)

 **See Also:**

"[Quick Start](#) (page 1-1)" and "[App Builder Concepts](#) (page 2-1)"

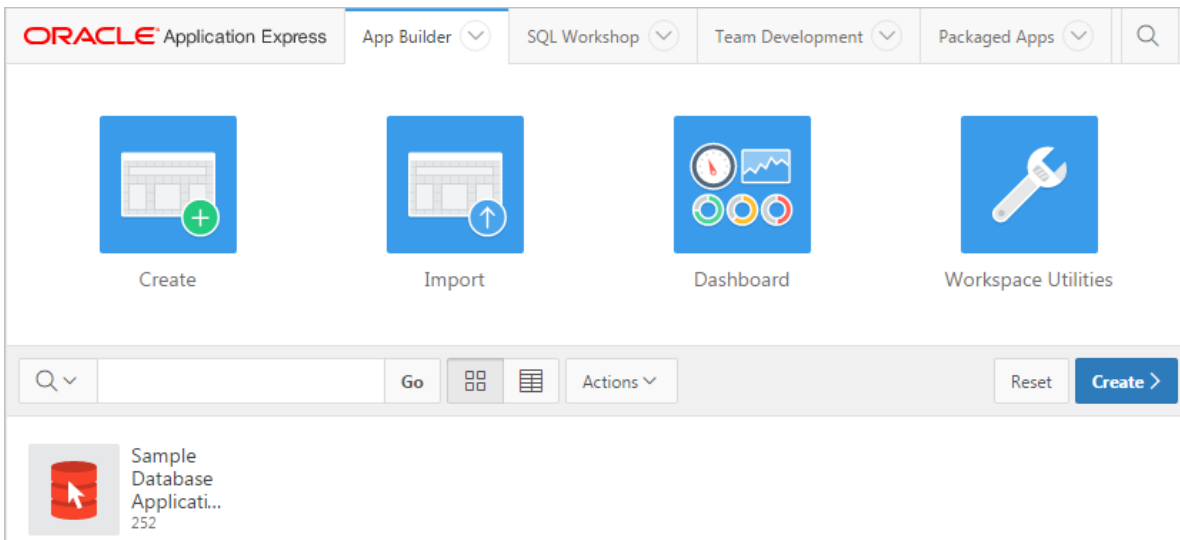
1.4.1 Accessing App Builder

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

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 Packaged Application** to access the Packaged Apps page and install a packaged application and load sample data. Once you create an application or install a packaged application, these regions no longer appear.

 **See Also:**

- ["Using the App Builder Home Page \(page 1-13\)"](#)
- ["Creating a Database Application \(page 6-4\)"](#)
- ["Installing a Packaged Application \(page 3-4\)"](#)

1.4.2 Using the App Builder Home Page

The App Builder home page displays all installed applications in the current Oracle Application Express instance. Developers use the App Builder home page to create new applications, select an application to edit, import previously exported applications, view the Dashboard, and 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 Packaged Application** to access the Packaged Apps page and install a packaged application and load sample data. Once you create an application or install a packaged application, these regions no longer appear.

- [App Builder Home Page \(page 1-13\)](#)
- [Reset and Create Buttons \(page 1-15\)](#)
- [App Builder Home Recent List \(page 1-16\)](#)
- [Migrations Link \(page 1-16\)](#)
- [Using Workspace Utilities \(page 1-16\)](#)

 **See Also:**

["Accessing App Builder \(page 1-12\)"](#) and ["Developer Navigation Tools \(page 1-6\)"](#)

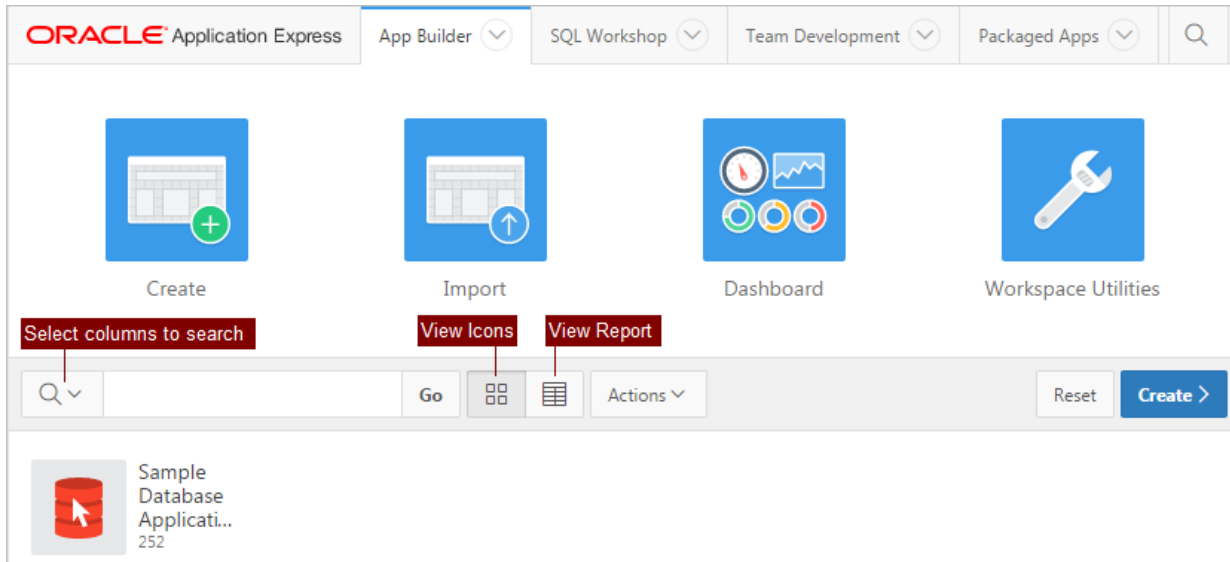
1.4.2.1 App Builder Home Page

App Builder Home Page Icons

The top of App Builder home page contain 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 displayed on the home page and Team Development home page, workspace themes, application groups, Application Express views, and application exports as well as access Cross Application Reports.



Applications and the Navigation Bar

The App Builder home page displays all installed applications in the current Oracle Application Express instance. You can customize the appearance of the App Builder home page 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.

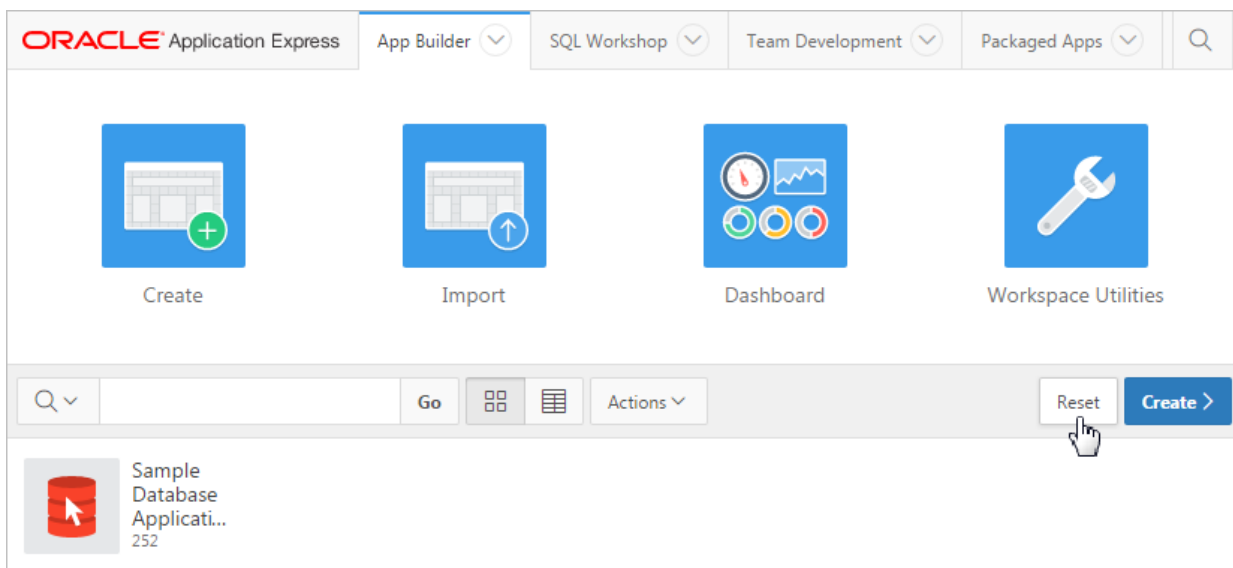
 **Tip:**

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

 **See Also:**

- ["Importing Export Files \(page 23-30\)"](#)
- ["Importing Export Files \(page 23-30\)"](#)
- ["Using Workspace Utilities \(page 1-16\)"](#)
- ["About the Actions Menu \(page 8-94\)"](#)

1.4.2.2 Reset and Create Buttons



The following buttons appear to the right of the navigation bar:

- **Reset.** Click **Reset** to return the page to the default display.
- **Create.** Click **Create** to create an application or install a packaged application.

1.4.2.3 App Builder Home Recent List

The **Recent** list displays on the right side of the page. This Region contains links to recently viewed applications.

1.4.2.4 Migrations Link

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

1.4.2.5 Using Workspace Utilities

Click the **Workspace Utilities** icon to access the Workspace Utilities page. The Workspace Utilities page contains links to common workspace utilities including App Builder Defaults, Manage News, Workspace Themes, Application Groups, Application Express Views, and the Export page.



See Also:

- [Leveraging App Builder Defaults in *Oracle Application Express Administration Guide*](#)
- [Managing News Entries](#) (page 4-28)
- [Managing Workspace Themes](#) (page 11-25)
- [Managing Application Groups](#) (page 6-71)
- [Accessing Application Express Views](#) (page 6-78)
- [Exporting an Application and Application Components](#) (page 23-16)
- [Accessing Cross Application Reports](#) (page 6-80)

1.4.3 Accessing the Application Home Page

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:**

[Using the Application Home Page](#) (page 1-17)

1.4.4 Using the Application Home Page

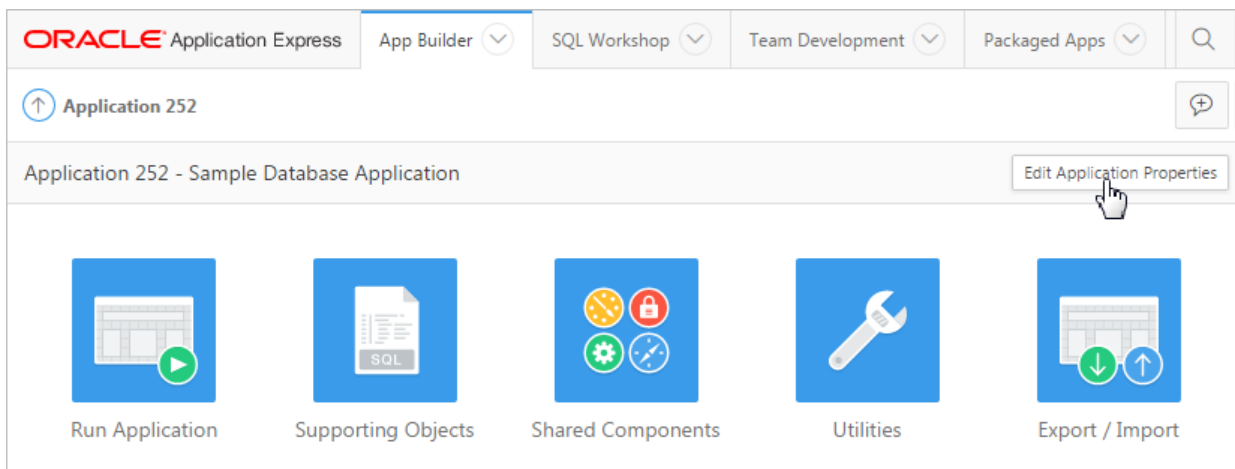
When a developer selects an application on the App Builder home page, the Application home page appears. Use the Application home page to run, edit, import, export, copy, or delete applications.

- [Edit Application Properties Button](#) (page 1-17)
- [Application Home Page Icons](#) (page 1-18)
- [Create Page Button](#) (page 1-19)
- [Application Home Page Navigation Bar](#) (page 1-20)
- [About Customizing the Application Home Page](#) (page 1-21)
- [Application Home Page Tasks List](#) (page 1-21)
- [Recently Edited Pages](#) (page 1-22)

 **See Also:**

["Developer Navigation Tools](#) (page 1-6)"

1.4.4.1 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.

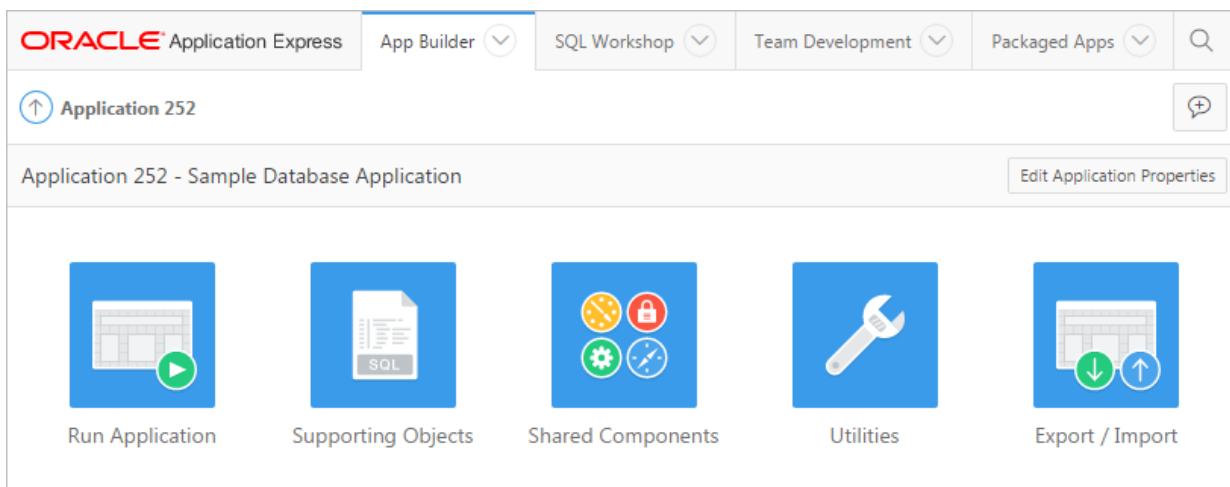
The Edit Application Properties button displays to the right of the application ID and name. Click **Edit Application Properties** to edit the application definition, including the application Name and Application Alias.



See Also:

["Editing Application Attributes \(page 6-16\)"](#)

1.4.4.2 Application Home Page Icons



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

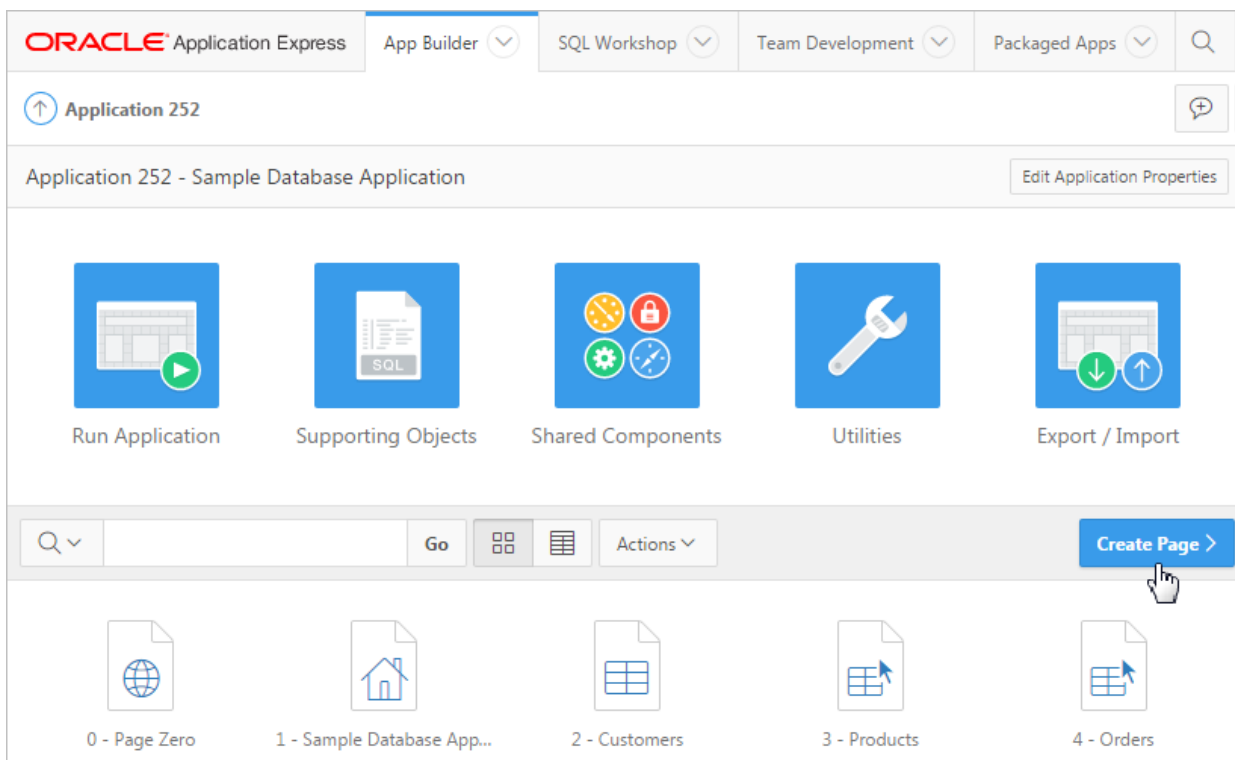
- **Run Application** submits the home page in the current application to the Application Express engine to render viewable HTML.
- **Supporting Objects** links to the Supporting Objects page. Use Supporting Objects to define database object installation scripts that are invoked when importing an application. 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:**

- ["How the Application Express Engine Renders and Processes Pages \(page 2-10\)"](#)
- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)
- ["Managing Shared Components \(page 16-1\)"](#)
- ["Using Application Utilities \(page 6-74\)"](#)
- ["Exporting an Application and Application Components \(page 23-16\)"](#)

1.4.4.3 Create Page Button



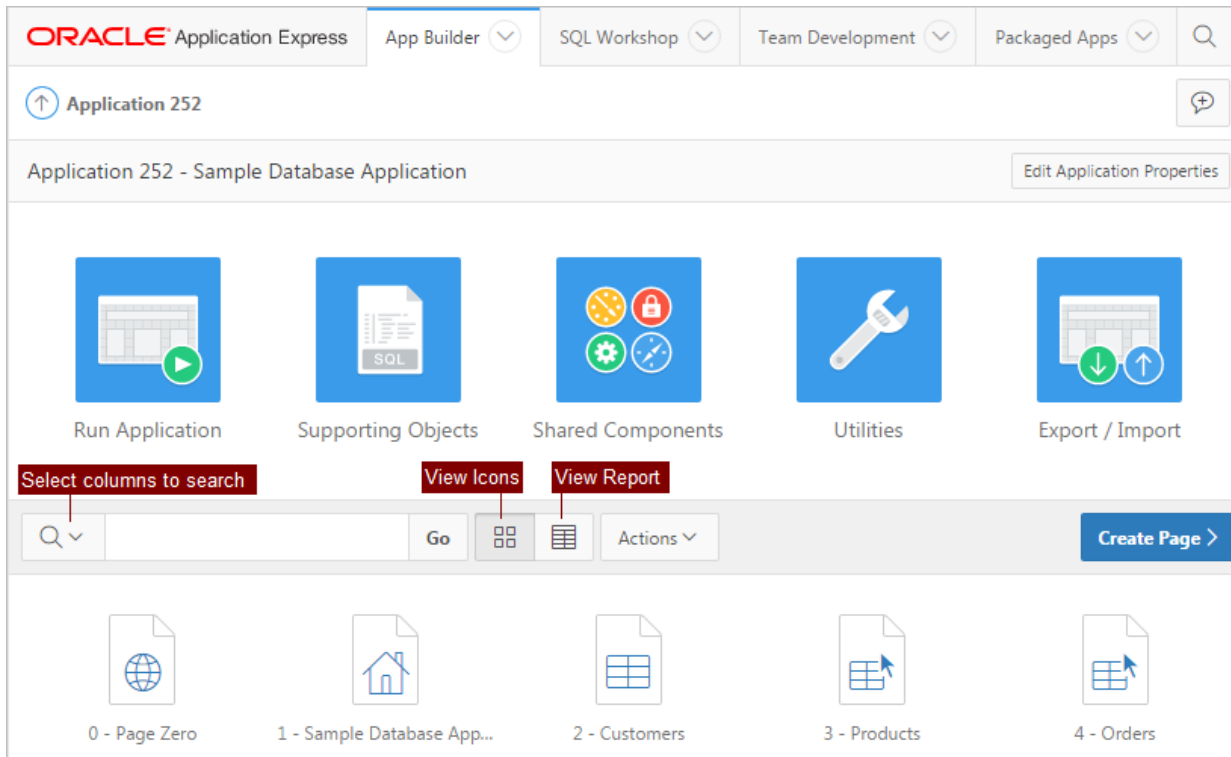
Clicking the **Create Page** button launches a wizard that walks you through creating a page in an application.

 **See Also:**

- ["Creating a Page from the Application Home Page \(page 6-48\)"](#)

1.4.4.4 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.
 - **View Report** displays each page as a line in a report. Each line includes the page ID, the page name, when the page was last updated, who updated it, the page type, user interface classification, and group.
- **Actions menu** - Displays the Actions menu. Use this menu to customize the report view.

 **See Also:**

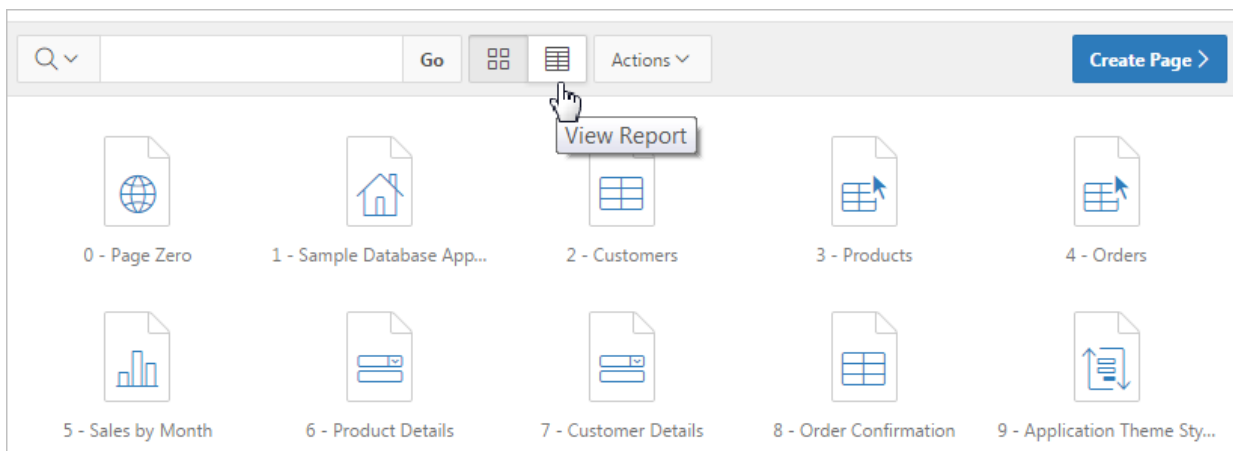
["App Builder Home Page \(page 1-13\)"](#)

1.4.4.5 About Customizing the Application Home Page

You can control how the Application home page appears by selecting the following icons:

- **View Icons** (the default) displays each page as an icon.
To view 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.

 **Tip:**

The Run icon does not display for dialog pages or global pages since these page cannot be run directly.

1.4.4.6 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 \(page 6-15\)"](#)
- ["Copying a Database Application \(page 6-9\)"](#)

1.4.4.7 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.

1.5 Editing Your Account Profile

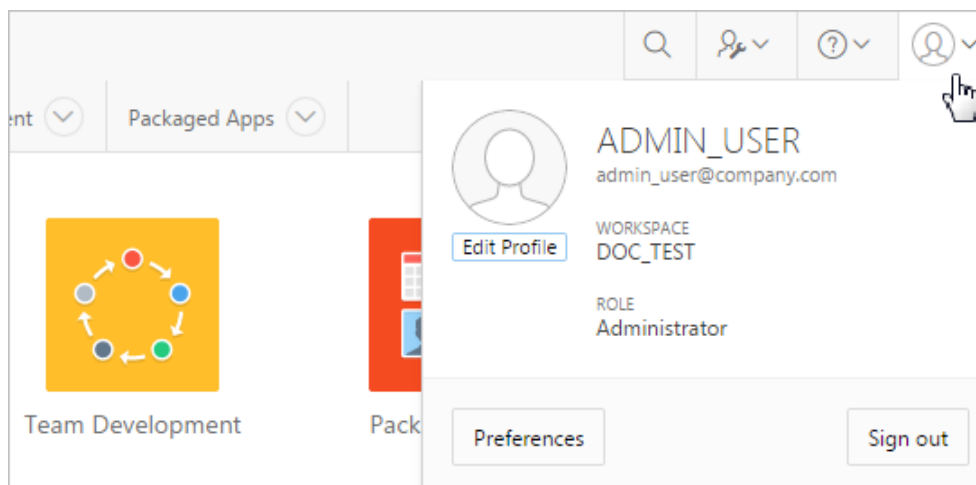
Oracle Application Express users have an account profile. You can edit your account profile to update your email address, edit your first or last name, upload a photo, or change your password.

- [Editing Your Account Profile \(page 1-22\)](#)
- [Changing Your Password \(page 1-23\)](#)

1.5.1 Editing Your Account Profile

To edit your account profile:

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



The Edit Profile dialog appears.

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

- Last Name

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

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

1.5.2 Changing Your Password

To reset your password:

1. Click the **Account** menu in the header region and select **Edit Profile**.
The Edit Profile dialog appears.
2. Scroll down to Change Password:
 - Enter Current Password - Enter your current password. Do not enter a password unless you want to change the current password.
 - Enter New Password - Enter a new password.
 - Confirm Password - Enter your new password again.
3. Click **Apply Changes**.

See Also:

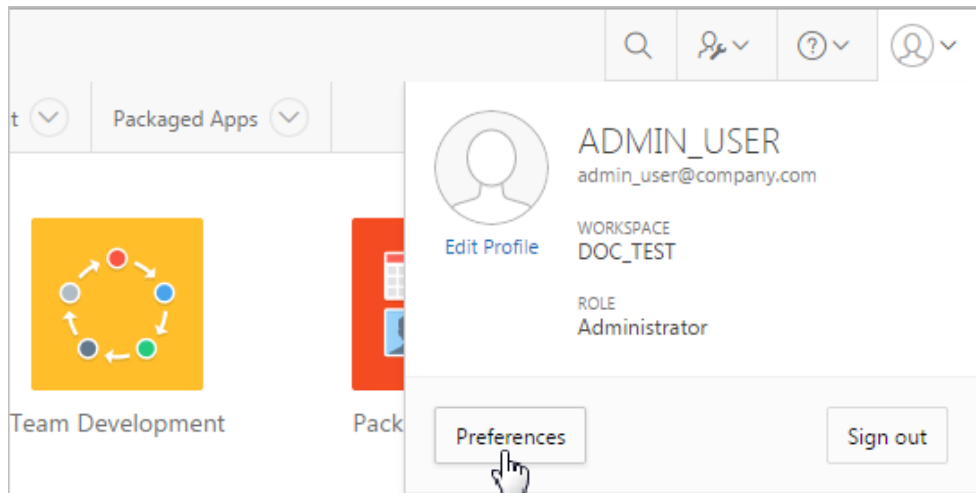
"Changing an End User Password" in *Oracle Application Express Administration Guide* and "[Account Menu](#) (page 1-10)"

1.6 Editing User Preferences

By editing your user preferences you can update the default workspace schema, specify a default date format, determine if Legacy Component View appears, and determine how Oracle Application Express manages windows and tabs when running applications.

To edit your user preferences:

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



The Preferences dialog appears.


2. 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.
- Enable Legacy Component View - Legacy Component View is disabled by default. Select **Yes** to enable Legacy Component View.
- 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**.

 **See Also:**
"Managing Pages in Legacy Component View (page 24-1)"

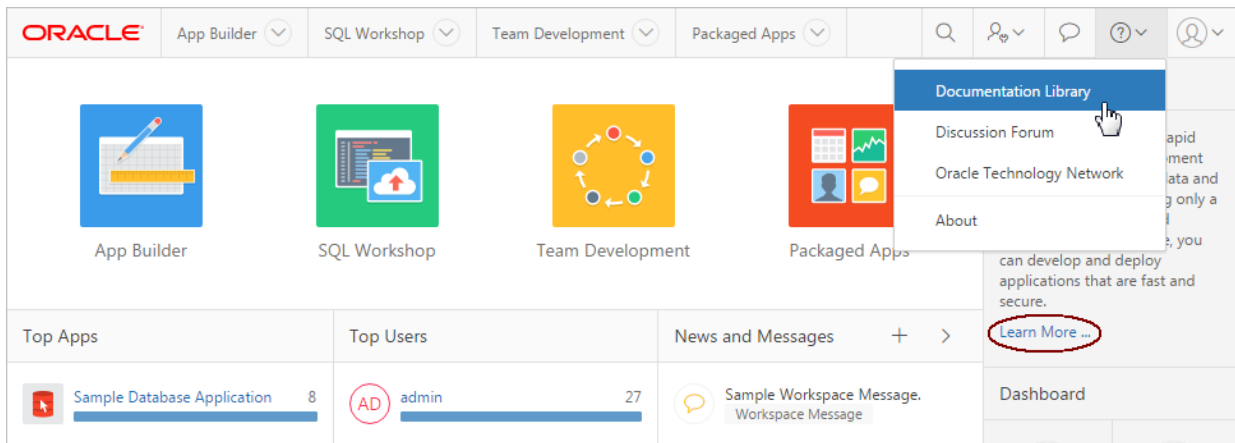
1.7 Using Oracle Application Express Documentation

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](#) (page 1-25)
- [Searching the Oracle Help Center](#) (page 1-26)
- [Downloading the Documentation Library](#) (page 1-26)
- [Viewing Field-Level Help](#) (page 1-27)

1.7.1 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.



To access Oracle Application Express documentation available in 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.
Information regarding the current page appears.

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*

1.7.2 Searching the Oracle Help Center

To perform a keyword search of the online documentation library:

1. Click the **Help** menu in the upper right corner of most Oracle Application Express pages.
2. Select **Documentation** .
Oracle Help Center appears.
3. 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.
4. 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.

1.7.3 Downloading the Documentation Library

Use the **Download all Oracle Application Express release 5.1 books** link to download the Oracle Application Express documentation library.

To download the online documentation library:

1. Access Oracle Help Center. Click the **Help** menu in the upper right corner and select **Documentation**

2. Under Download the Oracle Application Express Bookshelf, click **Download all Oracle Application Express release 5.1 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.

1.7.4 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](#) (page 1-27)
- [Viewing Help in Page Designer](#) (page 1-28)

See Also:

["Managing Pages in Page Designer](#) (page 7-1)" and ["Managing Pages in Legacy Component View](#) (page 24-1)"

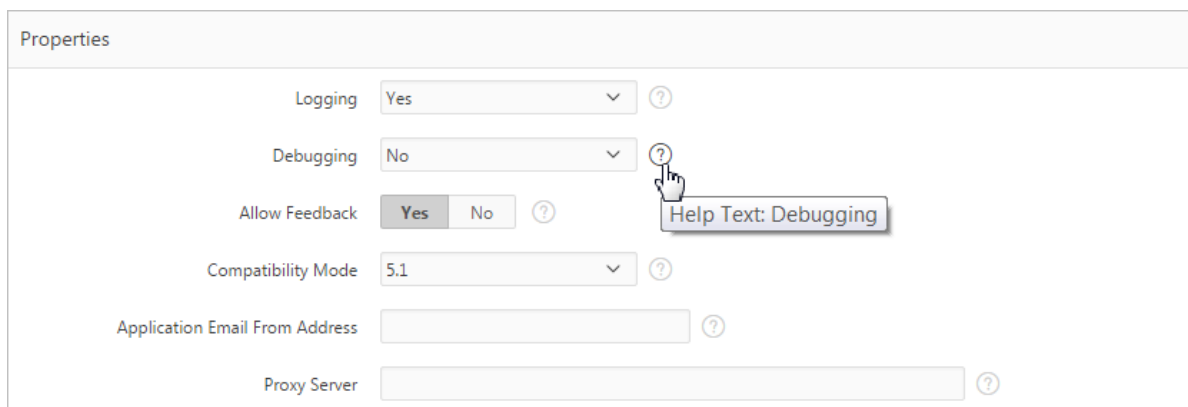
1.7.4.1 Viewing Field-level Help in Oracle Application Express

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 via the browser. Available options include:

- **Yes.** Enables the application to enable debug mode via the 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. However, for a production application, it is a good idea to disable debugging and thus prevent 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:**

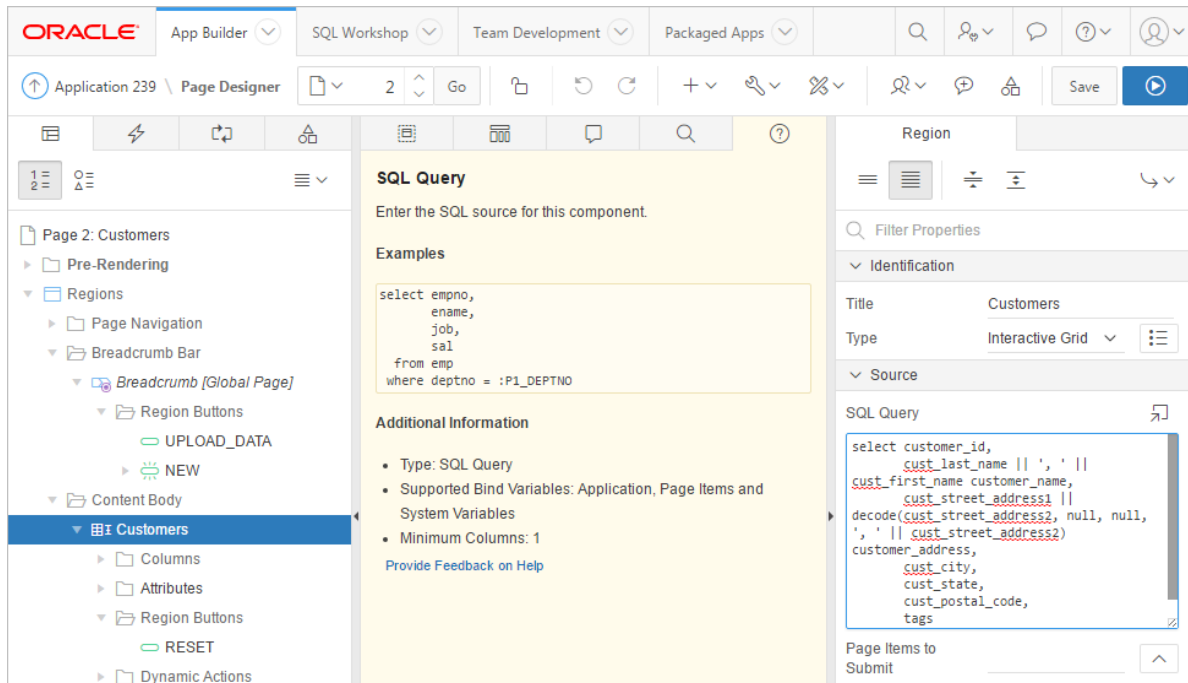
["Accessing Oracle Help Center \(page 1-25\)"](#)

1.7.4.2 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 \(page 7-22\)"](#)

2

App Builder Concepts

Developers use App Builder to build the pages that make up an Oracle Application Express application. Before using App Builder, developers need to understand some basic concepts about Application Express applications, viewing rendered application pages, managing session state, understanding URL syntax, and using built-in substitution strings.

Topics:

- [Understanding Oracle Application Express Applications](#) (page 2-1)
- [Managing User Interface Design](#) (page 2-5)
- [Running an Application or Page](#) (page 2-7)
- [How Does Page Processing and Page Rendering Work?](#) (page 2-9)
- [About Enabling Support for Bookmarks](#) (page 2-12)
- [Understanding Session State Management](#) (page 2-13)
- [Understanding URL Syntax](#) (page 2-16)
- [Managing Session State Values](#) (page 2-20)
- [Understanding Substitution Strings](#) (page 2-26)

See Also:

" [Understanding App Builder](#) (page 1-11)"

2.1 Understanding Oracle Application Express Applications

This section describes what an Oracle Application Express application is, explains the difference between database and Websheet applications, and how to run the Create Application Wizard.

- [What is an Oracle Application Express Application?](#) (page 2-2)
- [What Is an Application Page?](#) (page 2-2)
- [What is the Difference Between a Database and Websheet Application?](#) (page 2-2)
- [About Running the Create Application Wizard](#) (page 2-3)

**See Also:**

["Understanding App Builder \(page 1-11\)"](#)

2.1.1 What is an Oracle Application Express Application?

An 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.

2.1.2 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:**

["Managing Pages in Page Designer \(page 7-1\)"](#)

2.1.3 What is the Difference Between a Database and Websheet Application?

An Oracle Application Express application enables developers to manage and display data from 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. This section describes the differences between these application types.

- [About Database Applications \(page 2-2\)](#)
- [About Websheet Applications \(page 2-3\)](#)

2.1.3.1 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.

 **See Also:**

["Creating Database Applications \(page 6-1\)"](#)

2.1.3.2 About Websheet Applications

By creating Websheet applications, end users can 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 Websheet Applications \(page 5-1\)"](#)

2.1.4 About Running the Create Application Wizard

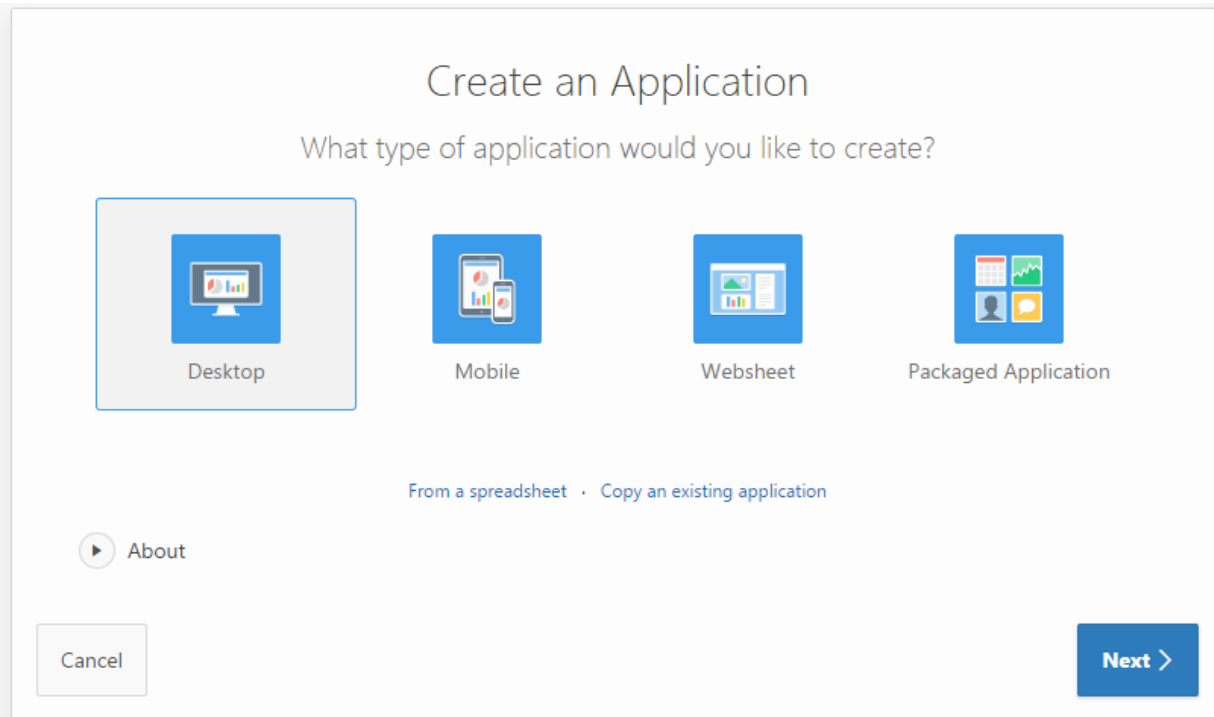
To create an Oracle Application Express application, a developer runs the Create Application Wizard. To run the Create Application Wizard, click the **Create** button on the App Builder home page.

 **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. Desktop and mobile database applications with both desktop and mobile user interfaces defined can utilize auto-detection to run the most appropriate user interface based on the screen size of the device. If the application is started from a personal computer then it will utilize the desktop user interface. However, if the application is run on a smartphone then it will utilize the

mobile user interface. In order to develop applications with multiple user interfaces, you must define the application with one user interface and then add the other from User Interface attributes.



Key decision points include:

- **Desktop.** Desktop database applications enable developers to fully control all aspects of development process and directly leverage their SQL and PL/SQL programming skills. Database applications use declarative control over flow control and support full user interface controls through the use of templates and themes. See "[Using the Create Application Wizard](#) (page 6-1)" and "[Creating a Database Application](#) (page 6-4)."
- **Mobile.** Mobile database applications are designed to run specifically on smartphone devices. See "[Using the Create Application Wizard](#) (page 6-1)" and "[Creating Mobile Applications](#) (page 6-11)."
- **Worksheet.** Worksheet applications enable users to build data centric applications without any SQL programming knowledge. Worksheet applications support pages, data grids, and reports and support community contributions. See "[Creating Worksheet Applications](#) (page 5-1)."
- **Packaged Application.** Links to the Packaged Apps gallery. See "[About Packaged Applications](#) (page 3-1)."
- Additional Create Application Options:
 - **From a spreadsheet.** Upload spreadsheet information as a file or by copying and pasting content. See "[Creating a Desktop Database Application from a Spreadsheet](#) (page 6-8)."
 - **Copy of existing application.** See "[Copying a Database Application](#) (page 6-9)."

**See Also:**

["Creating a Database Application \(page 6-4\)"](#)

2.2 Managing User Interface Design

When a developer first creates an application they select a user interface. App Builder supports multiple user interfaces within a single application.

- [About Determining Your Application User Interface \(page 2-5\)](#)
- [About Mobile Application Design \(page 2-5\)](#)
- [Understanding the Difference Between Mobile Design and Responsive Design \(page 2-6\)](#)

2.2.1 About Determining Your Application User Interface

App Builder supports multiple user interfaces within a single application. Developers can choose to build an application for the desktop, for a mobile device, or for both. Each user interface defined for an application has its own theme, login page, home page, and global page (formerly called Page 0). When you create a page within an application, you must specify which user interface is associated with that page. The user interface you select determines the options that display when creating pages and regions. For example, only desktop pages support maps, data loading, and interactive reports. Similarly, on mobile pages the most common construct is a list and form, instead of report and form.

**See Also:**

- ["Creating a Database Application \(page 6-4\)"](#)
- ["Creating Mobile Applications \(page 6-11\)"](#)
- ["Creating a Global Page to Display Components on Every Page \(page 12-3\)"](#)

2.2.2 About Mobile Application Design

Because mobile applications are designed to run specifically on smartphone devices, they must be built with extremely minimal, semantic HTML that is optimized for mobile connections. 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 the majority

of mobile devices. However, one major limitation is accessing on-device features such as contact lists. This limitation can be alleviated by integrating with solutions such as PhoneGap that 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/>



See Also:

"[Creating Mobile Applications](#) (page 6-11)" and "[About Responsive Design](#) (page 11-2)"

2.2.3 Understanding the Difference Between Mobile Design and Responsive Design

Oracle Application Express includes two approaches to developing applications for mobile devices:

- Option 1: Create jQuery Mobile-based web pages.
- Option 2: Use Theme 25 to create a responsive design in which the page layout fits the space regardless of the target device (for example, a desktop computer, laptop computer, tablet, or smartphone).

The main difference between these two approaches is that responsive design enables a single web page to display on any device. In contrast, developers building mobile applications with jQuery Mobile must create two sets of pages: one for desktop display and another for mobile devices. On the surface, creating a responsive design and having only a single page to maintain sounds more appealing than maintaining two separate pages. In practice, however, the up front cost of designing a truly responsive page are much higher and requires a much higher level of understanding of layout, HTML and CSS.

You should base your decision on whether to build separate mobile pages or create a responsive design based on the type of site or application you are developing. For example, applications that will be accessed by the general public, such as marketing sites, are well suited for responsive design. Productivity applications, such as customer tracker applications, are good candidates for building mobile pages where the native functionality of the smartphone can be utilized.



See Also:

"[Creating Mobile Applications](#) (page 6-11)" and "[About Responsive Design](#) (page 11-2)"

2.3 Running an Application or Page

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

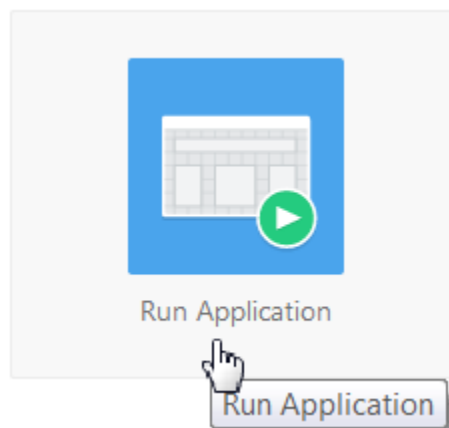
- [Run Application and Run Page Buttons](#) (page 2-7)
- [How Your Browser Impacts the Way Applications Run](#) (page 2-8)
- [Running an Application from the App Builder Home Page](#) (page 2-8)
- [Running an Application from the Application Home Page](#) (page 2-8)
- [Running a Page on the Application Home Page](#) (page 2-9)

See Also:

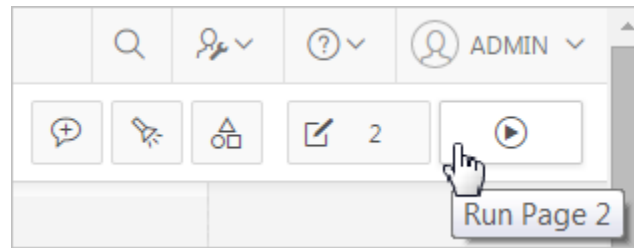
["Running a Page from Page Designer"](#) (page 7-37) and ["Running a Page in Legacy Component View"](#) (page 24-28)

2.3.1 Run Application and Run Page Buttons

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.



2.3.2 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 \(page 1-23\)"](#)

2.3.3 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:
 - Report view - Click **Run Page** icon in the far right column.
 - 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 \(page 2-8\)"](#)

2.3.4 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 \(page 2-8\)"](#)

2.3.5 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 \(page 6-64\)"](#) and ["How Your Browser Impacts the Way Applications Run \(page 2-8\)"](#)

2.4 How Does Page Processing and Page Rendering Work?

When you create an application in App Builder, you link pages together using tabs, navigation menus, buttons, or hypertext links. Each page can have buttons and items and can include application logic. You can branch from one page to the next using conditional navigation, perform calculations and validations, and display reports, calendars, and charts. You can generate reports, charts, and forms using built-in wizards, static HTML, or deliver more custom rendering with PL/SQL programming.

- [How the Application Express Engine Renders and Processes Pages \(page 2-10\)](#)
- [About Implicit Commit Points \(page 2-10\)](#)
- [Understanding Conditional Rendering and Processing \(page 2-10\)](#)
- [About Verifying User Identity \(page 2-12\)](#)

- [About Controlling Access to Controls and Components](#) (page 2-12)

2.4.1 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.

2.4.2 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*

2.4.3 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?](#) (page 2-11)
- [Selecting a Condition](#) (page 2-11)
- [Condition Example: Current Page in Expression 1](#) (page 2-11)
- [Condition Example: Exists](#) (page 2-11)
- [Condition Example: PL/SQL Expression](#) (page 2-12)

2.4.3.1 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](#) (page C-1)" for a detailed listing of available condition types

2.4.3.2 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.

2.4.3.3 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.

2.4.3.4 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](#) (page 2-24)"

2.4.3.5 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.

2.4.4 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 \(page 19-45\)"](#)

2.4.5 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 \(page 19-59\)"](#)

2.5 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](#) (page 19-7)" and "[Facilitating Bookmarks by Using Zero as the Session ID](#)" in *Oracle Application Express Application Builder User's Guide*, release 4.2

2.6 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?](#) (page 2-13)
- [What Is a Session?](#) (page 2-13)
- [About Session IDs](#) (page 2-14)
- [Viewing Session State](#) (page 2-14)

2.6.1 What is Session State?

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. In addition, 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.

2.6.2 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*

2.6.3 About Session IDs

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 (page 2-16)"

2.6.4 Viewing Session State

The behavior of an Oracle Application Express application is usually 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.

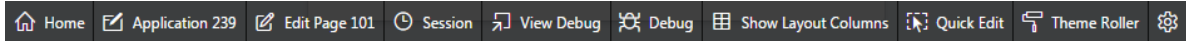
- [Viewing Session State for a Page](#) (page 2-14)
- [Using the Session Page](#) (page 2-15)

2.6.4.1 Viewing Session State for a Page

To view the session state for a page:

1. Run the application as a developer.

2. Locate the Developer toolbar at the bottom of the page.



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

The Session page appears.



See Also:

"[Using the Session Page](#) (page 2-15)"

2.6.4.2 Using the Session Page

The Session page provides valuable information about the session in which the application is currently running.

To control the display of the Session Page:

1. Page - Enter a page number.
2. Find - Enter a string to search for.
3. Rows - Select the number of rows to display.
4. View - Select the information to display.
5. Click **Set**.

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.

[Table 2-1](#) (page 2-15) describes the information that displays on the Session page depending upon the View option you select.

Table 2-1 View Options on the Session Page

View Option	Description
View, Page Items	Identify 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. See Also: " Understanding Page-Level Items (page 13-1)"

Table 2-1 (Cont.) View Options on the Session Page

View Option	Description
View, 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. See Also: " Managing Application-Level Items (page 16-20)" and " Understanding Substitution Strings (page 2-26)" for information about referencing item values
View, 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.
View, Collections	Displays currently defined collections. See Also: " About Using Collections (page 17-7)"

 **See Also:**

"[Viewing Session State for a Page](#) (page 2-14)"

2.7 Understanding URL Syntax

This section explains the URL for an Application Express application page, how to use f?p syntax to link pages, and how to call pages using a page alias or button URL.

- [About the Application Page URL](#) (page 2-16)
- [About Using f?p Syntax to Link Pages](#) (page 2-17)
- [Application Alias](#) (page 2-19)
- [Page Alias](#) (page 2-19)
- [Calling a Page Using an Application and Page Alias](#) (page 2-19)
- [Calling a Page from a Button URL](#) (page 2-20)

2.7.1 About the Application Page URL

The URL that displays for each page indicates the location of Oracle Application Express and identifies the address of Oracle Application Express, the application ID, page number, and session ID. For 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](#) (page 2-13)" and "[About Publishing the Database Application URL](#) (page 23-38)"

2.7.2 About Using f?p Syntax to Link Pages

You can create links between pages in your application using the following `f?p` syntax:

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

[Table 2-2](#) (page 2-17) describes the arguments you can pass when using `f?p` syntax.

Table 2-2 f?p Syntax Arguments

Syntax	Description
App	Indicates an application ID or alphanumeric application alias.
Page	Indicates a page number or alphanumeric page alias.
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: <code>&SESSION</code>. • PL/SQL: <code>V('SESSION')</code> • Bind variable: <code>:APP_SESSION</code>
Request	Sets the value of <code>REQUEST</code> . Each application button sets the value of <code>REQUEST</code> 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 <code>REQUEST</code> using the syntax: <ul style="list-style-type: none"> • Substitution string: <code>&REQUEST</code>. • PL/SQL: <code>V('REQUEST')</code> • Bind variable: <code>:REQUEST</code>

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

Syntax	Description
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.</p> <p>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 (page 22-1)"</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 (page 2-22)"</p>
itemNamees	<p>Comma-delimited list of item names used to set session state with a URL.</p>
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>
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>

It is important to understand how `f?p` syntax works. App Builder includes many wizards that automatically create these references for you. However, you may have to create the syntax yourself in some situations. The following sections describe specific instances that use `f?p` syntax to link pages.

2.7.3 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 \(page 6-16\)"](#) and ["Calling a Page Using an Application and Page Alias \(page 2-19\)"](#)

2.7.4 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 in Page Designer \(page 7-31\)"](#), ["Accessing Page Attributes in Legacy Component View \(page 24-12\)"](#), and ["Calling a Page Using an Application and Page Alias \(page 2-19\)"](#)

2.7.5 Calling a Page Using an Application and Page Alias

Application and page aliases must consist of valid Oracle identifiers, cannot contain any whitespace, and are not case-sensitive. The following example calls a page using an application and a page alias from within an application. It 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
```

See Also:

["Application Alias \(page 2-19\)"](#) and ["Page Alias \(page 2-19\)"](#)

2.7.6 Calling 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:

"[Creating Buttons](#) (page 13-51)" and "[APP_SESSION](#) (page 2-35)"

2.8 Managing Session State Values

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](#) (page 2-20)
- [Setting Session State](#) (page 2-21)
- [Clearing Session State](#) (page 2-22)
- [Referencing Session State Using Bind Variable Syntax](#) (page 2-24)
- [About Session Cloning](#) (page 2-25)



See Also:

"[Referencing Item Values](#) (page 13-2)"

2.8.1 About Referencing Session State

You can 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. [Table 2-3](#) (page 2-21) describes the supported syntax for referencing item values.

Table 2-3 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.

2.8.2 Setting Session State

This section describes how to set value of an item in your application and therefore set session state.

- [About Setting Session State](#) (page 2-21)
- [About Setting Session State with a Form Submission](#) (page 2-22)

2.8.2.1 About Setting Session State

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

- Form submission. See "[About Setting Session State with a Form Submission](#) (page 2-22)."
- Bind variable. See "[Referencing Session State Using Bind Variable Syntax](#) (page 2-24)."
- Computation. See "[Understanding Application Computations](#) (page 15-37)."
- f?p syntax. See "[About Using f?p Syntax to Link Pages](#) (page 2-17)."

2.8.2.2 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 2-3](#) (page 2-21).

2.8.3 Clearing Session State

As you develop your applications, you may find it useful to clear the cached value for specific items, all items on a page, all pages in an application, or the current user session. Clearing a cached value resets the value to null. This section provides specific examples of clearing session state.

- [About Clearing Cache for an Item](#) (page 2-22)
- [Clearing Cache for All Page Items](#) (page 2-22)
- [Example: Clearing Cache for an Entire Application](#) (page 2-24)
- [About Resetting an Application Completely](#) (page 2-24)
- [About Clearing Cache for the Current User Session](#) (page 2-24)

2.8.3.1 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.

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
```

2.8.3.2 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: Clearing Cache for Two Pages and Resetting Pagination](#) (page 2-23)
- [Example: Clearing Cache on a Page and Passing an Item Value](#) (page 2-23)

- [Example: Clearing Session Cache on a Page and Passing Values to Multiple Items](#) (page 2-23)

2.8.3.2.1 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).

2.8.3.2.2 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

2.8.3.2.3 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

2.8.3.3 Example: 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
```

2.8.3.4 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.

2.8.3.5 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
```

2.8.4 Referencing Session State Using Bind Variable Syntax

This section describes how to reference session state using bind variable syntax.

- [About Using Bind Variable Syntax](#) (page 2-24)
- [About Using Bind Variables in Regions Based on a SQL Query or LOV](#) (page 2-25)
- [About Using Bind Variables in Regions Based on PL/SQL](#) (page 2-25)

2.8.4.1 About Using Bind Variable Syntax

You can use bind variables syntax anywhere in Oracle Application Express where you are using SQL or PL/SQL to reference session state of a specified item. For example:

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

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

2.8.4.2 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 (plssql 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 \(page 12-4\)"](#) for information about creating regions

2.8.4.3 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`.

2.8.5 About Session Cloning

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.

Developers can now 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. To accomplish this, 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
```

When using this feature, developer 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

2.9 Understanding 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](#) (page 2-26)
- [Controlling Output Escaping in Substitution Strings](#) (page 2-27)
- [About Determining Substitution String Usage within Templates](#) (page 2-28)
- [Using Built-in Substitution Strings](#) (page 2-28)

2.9.1 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#
```

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.
```

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

2.9.2 Controlling Output Escaping in Substitution Strings

Developers can append an exclamation mark (!) followed by a predefined filter name to a page or application item name, report column, or other 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. 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 include:

- HTML escapes reserved HTML characters, for example:

```
&ITEM!HTML.  
#REPORT_COLUMN!HTML#
```

See Also:

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

- ATTR escapes reserved characters in a HTML attribute context, for example:

```
#REPORT_COLUMN!ATTR#
```

See Also:

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

- JS escapes reserved characters in a JavaScript context, for example:

```
&ITEM!JS.  
#REPORT_COLUMN!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!RAW.  
#REPORT_COLUMN!RAW#
```
- `STRIPHTML` removes HTML tags from the output and escapes reserved HTML characters, for example:

```
&ITEM!STRIPHTML.  
#REPORT_COLUMN!STRIPHTML#
```

2.9.3 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](#) (page 11-32)."

2.9.4 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](#) (page 2-29)
- [APEX\\$ROW_SELECTOR](#) (page 2-30)
- [APEX\\$ROW_STATUS](#) (page 2-30)
- [APP_ID](#) (page 2-30)
- [APP_ALIAS](#) (page 2-31)
- [APP_AJAX_X01, ... APP_AJAX_X10](#) (page 2-31)
- [APP_DATE_TIME_FORMAT](#) (page 2-32)
- [APP_IMAGES](#) (page 2-32)
- [APP-NLS_DATE_FORMAT](#) (page 2-33)
- [APP-NLS_TIMESTAMP_FORMAT](#) (page 2-33)
- [APP-NLS_TIMESTAMP_TZ_FORMAT](#) (page 2-34)
- [APP_PAGE_ALIAS](#) (page 2-34)
- [APP_PAGE_ID](#) (page 2-34)
- [APP_REQUEST_DATA_HASH](#) (page 2-35)
- [APP_SESSION](#) (page 2-35)
- [APP_SESSION_VISIBLE](#) (page 2-36)

- [APP_TITLE](#) (page 2-36)
- [APP_UNIQUE_PAGE_ID](#) (page 2-37)
- [APP_USER](#) (page 2-37)
- [AUTHENTICATED_URL_PREFIX](#) (page 2-38)
- [BROWSER_LANGUAGE](#) (page 2-39)
- [CURRENT_PARENT_TAB_TEXT](#) (page 2-39)
- [DEBUG](#) (page 2-39)
- [HOME_LINK](#) (page 2-40)
- [IMAGE_PREFIX](#) (page 2-40)
- [JET_BASE_DIRECTORY](#) (page 2-41)
- [JET_CSS_DIRECTORY](#) (page 2-41)
- [JET_JS_DIRECTORY](#) (page 2-41)
- [LOGIN_URL](#) (page 2-41)
- [LOGOUT_URL](#) (page 2-42)
- [PRINTER_FRIENDLY](#) (page 2-42)
- [PROXY_SERVER](#) (page 2-42)
- [PUBLIC_URL_PREFIX](#) (page 2-43)
- [REQUEST](#) (page 2-43)
- [Using REQUEST](#) (page 2-43)
- [SCHEMA_OWNER](#) (page 2-45)
- [SQLERRM](#) (page 2-45)
- [SYSDATE_YYYYMMDD](#) (page 2-45)
- [THEME_DB_IMAGES](#) (page 2-46)
- [THEME_IMAGES](#) (page 2-46)
- [WORKSPACE_IMAGES](#) (page 2-46)
- [WORKSPACE_ID](#) (page 2-47)

 **See Also:**

"[Substitutions](#) (page 6-23)" and "[Establishing User Identity Through Authentication](#) (page 19-45)"

2.9.4.1 APEX\$ROW_NUM

`APEX$ROW_NUM` refers the currently processed row number of a submitted tabular form data. You can use this placeholder in validations, processes, and conditions associated with a tabular form to refer to the row number of the currently processed tabular form row. [Table 2-4](#) (page 2-30) describes the supported syntax for referencing `APEX$ROW_NUM`.

Table 2-4 APEX\$ROW_NUM

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

2.9.4.2 APEX\$ROW_SELECTOR

Use `APEX$ROW_SELECTOR` in validations, processes, and conditions associated with a tabular form to refer to the row selector check box in a tabular form. This placeholder returns `X` if the tabular form row selector check box of the currently processed tabular form row is checked and `NULL` if it unchecked. [Table 2-5](#) (page 2-30) describes the supported syntax for referencing `APEX$ROW_SELECTOR`.

Table 2-5 APEX\$ROW_SELECTOR

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

2.9.4.3 APEX\$ROW_STATUS

Use `APEX$STATUS` in validations, processes, and conditions associated with a tabular form to refer to the row status in a tabular form. This placeholder returns the status of `C` if created, `U` if updated, or `D` if deleted for the currently processed tabular form row. [Table 2-6](#) (page 2-30) describes the supported syntax for referencing `APEX$ROW_STATUS`.

Table 2-6 APEX\$ROW_STATUS

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

2.9.4.4 APP_ID

`APP_ID` identifies the application ID of the currently executing application. [Table 2-7](#) (page 2-30) describes the supported syntax for referencing `APP_ID`.

Table 2-7 APP_ID Syntax

Reference Type	Syntax
Bind variable	:APP_ID

Table 2-7 (Cont.) APP_ID Syntax

Reference Type	Syntax
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.
```

2.9.4.5 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 2-8](#) (page 2-31) describes the supported syntax for referencing APP_ALIAS.

Table 2-8 APP_ALIAS Syntax

Reference Type	Syntax
Bind variable	:APP_ALIAS
PL/SQL	V('APP_ALIAS')
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>
```

2.9.4.6 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](#) (page 2-32) describes the supported syntax for referencing APP_AJAX_Xnn.

Table 2-9 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

2.9.4.7 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](#) (page 2-32) 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.

2.9.4.8 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](#) (page 2-32) 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.

Table 2-11 (Cont.) APP_IMAGES Syntax

Reference Type	Syntax
Template substitution	#APP_IMAGES#

 **See Also:**

"[IMAGE_PREFIX](#) (page 2-40)," "[WORKSPACE_IMAGES](#) (page 2-46)," and "[About Managing Images](#) (page 12-31)"

2.9.4.9 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](#) (page 2-33) 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.

2.9.4.10 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](#) (page 2-33) 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.

2.9.4.11 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](#) (page 2-34) 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.

2.9.4.12 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](#) (page 2-34) 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.

2.9.4.13 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](#) (page 2-34) describes the supported syntax for referencing APP_PAGE_ID.

Table 2-16 APP_PAGE_ID Syntax

Reference Type	Syntax
Bind variable	:APP_PAGE_ID

Table 2-16 (Cont.) APP_PAGE_ID Syntax

Reference Type	Syntax
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.
```

2.9.4.14 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](#) (page 2-35) 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.

2.9.4.15 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](#) (page 2-35) 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.

2.9.4.16 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 (page 2-36) 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;
```

2.9.4.17 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 (page 2-36) describes the supported syntax for referencing APP_TITLE.

Table 2-20 APP_TITLE Syntax

Reference Type	Syntax
Bind variable	:APP_TITLE

Table 2-20 (Cont.) APP_TITLE Syntax

Reference Type	Syntax
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.
```

2.9.4.18 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](#) (page 2-37) 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.

2.9.4.19 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](#) (page 2-38) 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](#) (page 6-24)" for information about the Public User attribute

2.9.4.20 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](#) (page 2-38) 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.

2.9.4.21 BROWSER_LANGUAGE

`BROWSER_LANGUAGE` refers to the web browser's current language preference. [Table 2-24](#) (page 2-39) 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.

2.9.4.22 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](#) (page 2-39) 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.

2.9.4.23 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](#) (page 2-39) 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
```

2.9.4.24 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](#) (page 2-40) describes the supported syntax for referencing `HOME_LINK`.

Table 2-27 HOME_LINK Syntax

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



See Also:

["Managing the Application User Interface \(page 6-36\)"](#) and ["User Interface Details Page \(page 6-40\)"](#) for information about the Home URL attribute

2.9.4.25 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](#) (page 2-40) 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 \(page 2-32\)"](#)
- ["WORKSPACE_IMAGES \(page 2-46\)"](#)
- ["Editing Application Attributes \(page 6-16\)"](#)

2.9.4.26 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#
```

2.9.4.27 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#
```

2.9.4.28 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#
```

2.9.4.29 LOGIN_URL

Use `LOGIN_URL` to display a link to a login page for users that are not currently logged in. [Table 2-29 \(page 2-41\)](#) describes the supported syntax for `LOGIN_URL`.

 **See Also:**

["Authentication \(page 6-24\)"](#) and ["Security Attributes Page \(page 6-24\)"](#)

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.

Table 2-29 (Cont.) LOGIN_URL Syntax

Reference Type	Syntax
Template Substitution	#LOGIN_URL#

2.9.4.30 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](#) (page 2-42) 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#

2.9.4.31 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 2-31](#) (page 2-42) describes the supported syntax for referencing PRINTER_FRIENDLY.

Table 2-31 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.

2.9.4.32 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
```

2.9.4.33 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 2-32](#) (page 2-43) describes the supported syntax for referencing `PUBLIC_URL_PREFIX`.

Table 2-32 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#

2.9.4.34 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 2-33](#) (page 2-43) describes the supported syntax for referencing `REQUEST`.

Table 2-33 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" \(page 2-43\)](#)

2.9.4.35 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](#) (page 2-44)
- [About the When Button Pressed Attribute](#) (page 2-44)

- [About Referencing REQUEST Using Declarative Conditions](#) (page 2-44)
- [About Using REQUEST for Show Processing](#) (page 2-44)
- [About Using BRANCH_TO_PAGE_ACCEPT](#) (page 2-45)

**See Also:**

"REQUEST (page 2-43)"

2.9.4.35.1 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`.

2.9.4.35.2 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.

2.9.4.35.3 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**.

2.9.4.35.4 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

2.9.4.35.5 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.

2.9.4.36 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).

2.9.4.37 SQLERRM

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

```
#SQLERRM#
```

2.9.4.38 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 2-34 SYSDATE_YYYYMMDD Syntax

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

2.9.4.39 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#
```

2.9.4.40 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#
```

2.9.4.41 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 2-35](#) (page 2-46) describes the supported syntax for referencing `WORKSPACE_IMAGES`.

Table 2-35 WORKSPACE_IMAGES Syntax

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



See Also:

"[APP_IMAGES](#) (page 2-32)" and "[IMAGE_PREFIX](#) (page 2-40)"

2.9.4.42 WORKSPACE_ID

Use this substitution string to reference the workspace ID. [Table 2-36](#) (page 2-47) describes the supported syntax for referencing `WORKSPACE_ID`.

Table 2-36 `WORKSPACE_ID` Syntax

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

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 Packaged Applications

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

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 Worksheet Objects in New Workspaces" in *Oracle Application Express Administration Guide*

Topics:

- [About Packaged Applications](#) (page 3-1)
- [Managing Packaged Applications in a Full Development Environment](#) (page 3-2)
- [Understanding Sample Database Application](#) (page 3-8)
- [Understanding AnyCo IT Department Worksheet](#) (page 3-19)
- [Utilizing Packaged App Administration](#) (page 3-31)
- [Using a Packaged Apps Only Workspace](#) (page 3-35)

3.1 About Packaged Applications

The Packaged Application 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 packaged applications before you can edit them. Unlocking an application makes it ineligible for future upgrades or support by Oracle Support.

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.

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 be editable by default.

Custom Applications

Custom applications are applications you create and that are managed by your instance administrator. Like Oracle-supplied packaged applications, custom packaged 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*

3.2 Managing Packaged Applications in a Full Development Environment

In a full development environment, developers can view, install, run, remove, unlock, export, change authentication, and update packaged applications.

- [Viewing Packaged Apps](#) (page 3-3)
- [Installing a Packaged Application](#) (page 3-4)
- [Running an Installed Packaged Application](#) (page 3-5)
- [Removing an Installed Packaged Application](#) (page 3-6)
- [Changing Authentication for an Installed Packaged Application](#) (page 3-6)
- [Unlocking an Installed Productivity Application](#) (page 3-7)
- [Updating a Packaged Application](#) (page 3-7)



See Also:

"Managing Custom Applications" in *Oracle Application Express Administration Guide* and "[Using a Packaged Apps Only Workspace](#) (page 3-35)"

3.2.1 Viewing Packaged Apps

Use the Packaged App page to view, install, run, remove, unlock, export, change authentication, and update packaged applications.

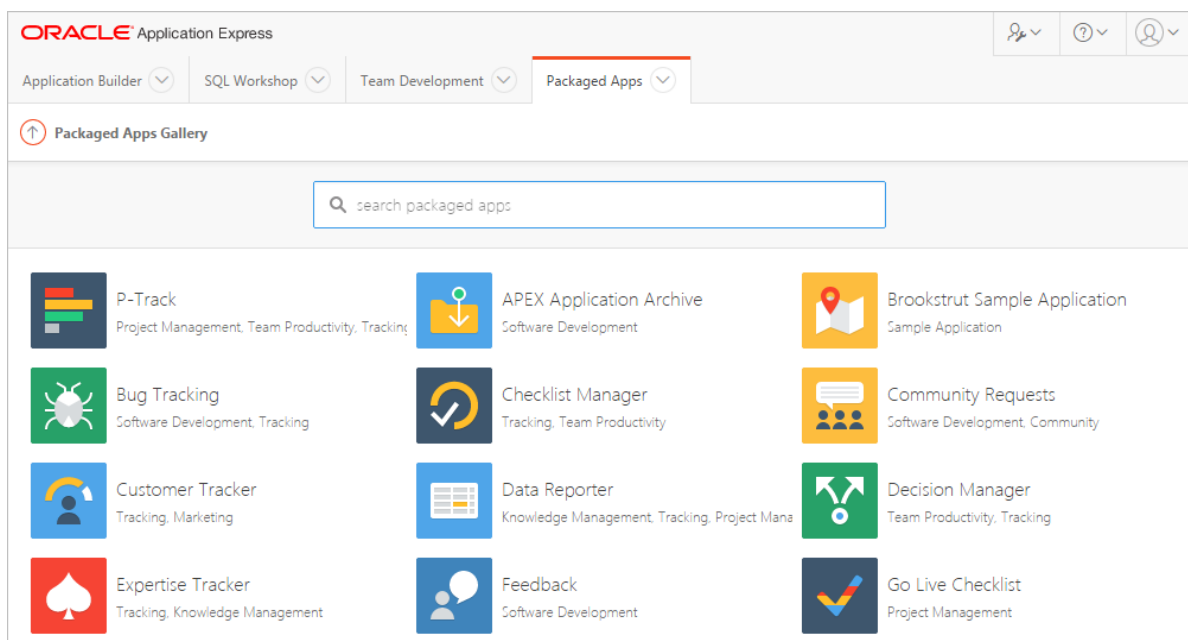
 **Tip:**

In new workspace, a gray box displays on the **App Builder** home page labeled, **Install a Packaged Application**. Click **Install a Packaged Application** as a shortcut to access the Packaged Apps page to install a packaged application.

To access the Packaged App page:

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

The Packaged Apps page appears.



3. Locate an application to install. Use the Search field at the top of the page to narrow the display.

The Packaged Application Details page appears, displaying a summary information about the application.

4. On Packaged Application Details page, you can:



Tip:

The actual options that display depend upon whether the application is installed.

- **Install Packaged App.** If not installed, click **Install Packaged 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.



See Also:

- ["Installing a Packaged Application \(page 3-4\)"](#)
- ["Removing an Installed Packaged Application \(page 3-6\)"](#)
- ["Changing Authentication for an Installed Packaged Application \(page 3-6\)"](#)
- ["Unlocking an Installed Productivity Application \(page 3-7\)"](#)
- ["Updating a Packaged Application \(page 3-7\)"](#)

3.2.2 Installing a Packaged Application



Tip:

In a new workspace, a gray box displays on the **App Builder** home page labeled, **Install a Packaged Application**. Click **Install a Packaged Application** as a shortcut to access the Packaged Apps page.

To install a packaged application:

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

7. Click **Install Packaged App** again.

The Packaged Application Details 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 Packaged Apps Only Workspace.

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

 **See Also:**


- ["Changing Authentication for an Installed Packaged Application \(page 3-6\)"](#)
- ["Unlocking an Installed Productivity Application \(page 3-7\)"](#)
- ["Removing an Installed Packaged Application \(page 3-6\)"](#)
- ["Running an Installed Packaged Application \(page 3-5\)"](#)

3.2.3 Running an Installed Packaged Application

To run an installed packaged application:

1. On the Workspace home page, click the **Packaged Apps** icon.
The Packaged Apps 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 \(page 2-8\)"](#)

3.2.4 Removing an Installed Packaged Application

To remove an installed packaged application:

1. On the Workspace home page, click the **Packaged Apps** icon.
The Packaged Apps 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 "[Viewing Packaged Apps](#) (page 3-3)."

3.2.5 Changing Authentication for an Installed Packaged Application

To change the authentication for an installed packaged application:

1. On the Workspace home page, click the **Packaged Apps** icon.
The Packaged Apps 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 "[Viewing Packaged Apps](#) (page 3-3)."

 **See Also:**

"[Establishing User Identity Through Authentication](#) (page 19-45)"

3.2.6 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 **Packaged Apps** icon.
The Packaged Apps page appears.
2. Click the **Installed** tab and select the application.
3. Click **Manage**.
4. Click **Unlock**.

 **Tip:**

Unlock is NOT available in a Packaged Apps Only Workspace.

The Unlock Packaged Application page appears.

5. Click **Unlock Application**.

 **See Also:**

["Installing a Packaged Application \(page 3-4\)"](#)

3.2.7 Updating a Packaged Application

You can update a packaged application if an Update button displays on Packaged Application Details page.

To update a packaged application:

1. On the Workspace home page, click the **Packaged Apps** icon.
The Packaged Apps page appears.
2. Click the **Installed** tab and select the application to be removed.
3. Click **Update Application**.
The Update Packaged 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 Packaged Apps gallery.

 **See Also:**

["Running an Installed Packaged Application \(page 3-5\)"](#)

3.3 Understanding Sample Database Application

Sample Database Application features an easy-to-use interface for viewing, updating, and searching order and customer information for clothing products. Users can navigate among the pages using the Home, Customers, Products, Orders, and Reports navigation menu. Like all sample packaged applications, *Sample Database Application* is fully editable.

- [Installing and Running Sample Database Application \(page 3-8\)](#)
- [Understanding Sample Database Application \(page 3-9\)](#)
- [Modifying Sample Database Application \(page 3-17\)](#)

3.3.1 Installing and Running Sample Database Application

To install *Sample Database Application*:

1. Sign in to Oracle Application Express.
The Workspace home page appears.
2. Click the **Packaged Apps** icon.
The Packaged Apps 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 Packaged App Details page appears.

4. Click **Install Packaged App**.
5. Select an Authentication scheme and click **Next**.
6. Click **Install Packaged App** again.

The Packaged Application Details 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 packaged application, it displays under Installed Packaged Apps on the Packaged Apps page.

3.3.2 Understanding Sample Database Application

Sample Database Application demonstrates:

- Examples of ways to display summary information.
- Reports and forms for viewing, updating, and adding customers, products, and orders
- Various charts and maps available in Oracle Application Express
- Mobile pages to replicate the majority of functionality available in the desktop pages

Sample Database Application features the following built-in functionality and key pages:

 **Tip:**

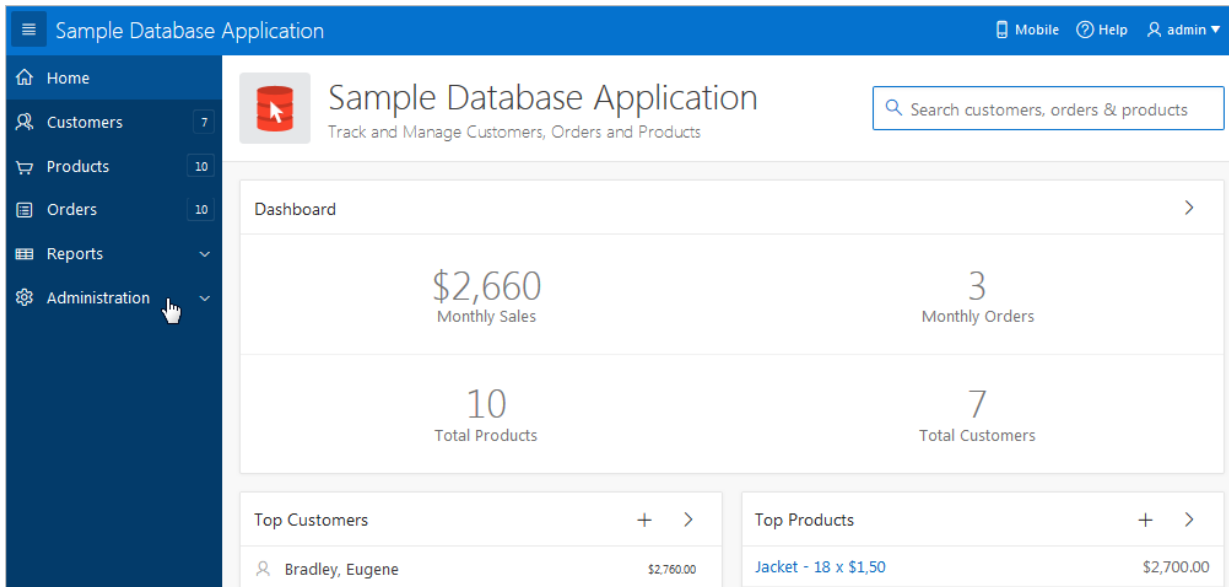
The size of left navigation menu changes depending upon the size of the window. In some of the images in this section, the navigation menu is minimized.

- [Home Page](#) (page 3-10)
- [Customers Page](#) (page 3-11)
- [Using the Customers Page](#) (page 3-11)
- [Products Page](#) (page 3-12)
- [Using the Products Page](#) (page 3-13)
- [Orders Page](#) (page 3-14)
- [Using the Orders Page](#) (page 3-14)
- [Reports Page](#) (page 3-15)
- [Administration Page](#) (page 3-15)
- [Accessing the Mobile Version](#) (page 3-16)

 **See Also:**

"What Is an Application Page? (page 2-2)"

3.3.2.1 Home Page



The home page contains the following main regions:

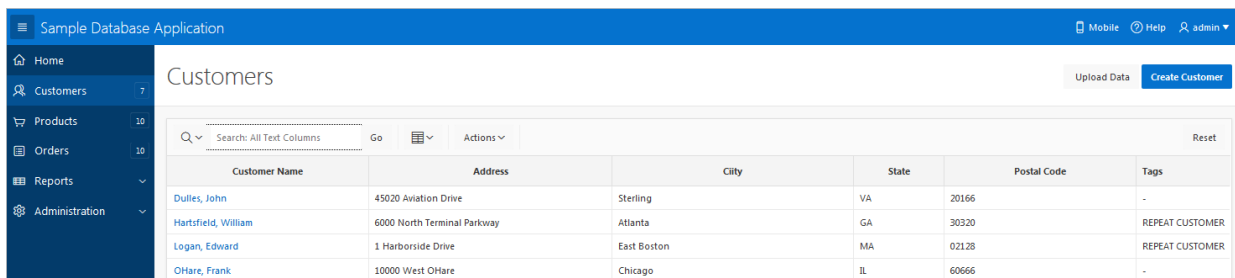
- **Dashboard** - Demonstrates the use of a Badge List plug-in. This region displays a value based on an underlying SQL statement. Links in this region take you to reports on Monthly Sales, Monthly Orders, Total Products, and Total Customers.
- **Top Customers** - Demonstrates the Slider Tooltip plug-in. 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 \(page 8-1\)"](#)
- ["Managing Regions in Page Designer \(page 12-5\)"](#)
- ["Creating Lists \(page 16-27\)"](#)

3.3.2.2 Customers Page

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



Customer Name	Address	City	State	Postal Code	Tags
Dulles, John	45020 Aviation Drive	Sterling	VA	20166	-
Hartsfield, William	6000 North Terminal Parkway	Atlanta	GA	30320	REPEAT CUSTOMER
Logan, Edward	1 Harborside Drive	East Boston	MA	02128	REPEAT CUSTOMER
O'Hare, Frank	10000 West O'Hare	Chicago	IL	60666	-

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 \(page 8-13\)"](#)

3.3.2.3 Using the Customers Page

This section describes functionality built into the Customers page.

- [Editing a Customer \(page 3-11\)](#)
- [Creating a Customer \(page 3-12\)](#)
- [Uploading Data \(page 3-12\)](#)

3.3.2.3.1 Editing a Customer

To update existing customer information:

1. Click the customer name.
2. On the Customer Details page, edit the fields and click **Apply Changes**.
3. To delete a customer, click **Delete**.

3.3.2.3.2 Creating a Customer

To add a new customer:

1. Click the **Create Customer** button.
2. On the Customer Details page, edit the fields and click **Add Customer**.

3.3.2.3.3 Uploading Data

To upload data:

1. Click the **Upload Data** button.
2. For Import From:
 - a. Import From - Select either:
 - Upload file, comma separated (*.csv) or tab delimited. If selected, select the file in the **File Name** form element.
 - Copy and Paste. If selected, a Copy and Paste field displays at the bottom on the window.
 - 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. File Character Set - Select the appropriate file character set.
 - f. Click **Next**.
3. On Data/Table Mapping, review the mappings and click **Next**.
4. Click **Load Data**.
5. Review Results.
6. Click **Finish**.

3.3.2.4 Products Page

The Products page enables users to view and edit product information. The Products region is 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

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](#) (page 8-92)" and
 "[About the Actions Menu](#) (page 8-94)"

3.3.2.5 Using the Products Page

This section describes functionality built into the Customers page.

- [Editing a Product](#) (page 3-13)
- [Adding a Product](#) (page 3-13)

3.3.2.5.1 Editing a Product

To update existing product:

1. Click the product name.
2. On the Product Details page, edit the fields and click **Apply Changes**.
3. To delete a product, click **Delete**.

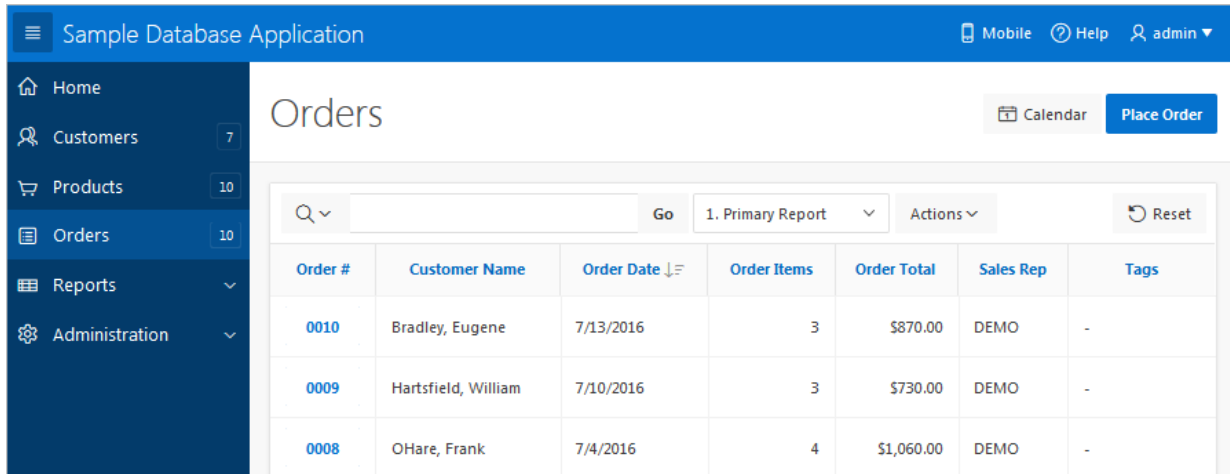
3.3.2.5.2 Adding a Product

To add a new product:

1. Click the **Add Product** button.
2. On the Product Details page, edit the fields and click **Add Product**.

3.3.2.6 Orders Page

The Orders page enables users to create, view, and edit customer orders.



Order #	Customer Name	Order Date ↓	Order Items	Order Total	Sales Rep	Tags
0010	Bradley, Eugene	7/13/2016	3	\$870.00	DEMO	-
0009	Hartsfield, William	7/10/2016	3	\$730.00	DEMO	-
0008	OHare, Frank	7/4/2016	4	\$1,060.00	DEMO	-

Orders 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.



See Also:

["Managing Interactive Reports \(page 8-38\)"](#)

3.3.2.7 Using the Orders Page

This section describes functionality built into the Orders page.

- [Editing an Order](#) (page 3-14)
- [Placing an Order](#) (page 3-14)

3.3.2.7.1 Editing an Order

To edit an order:

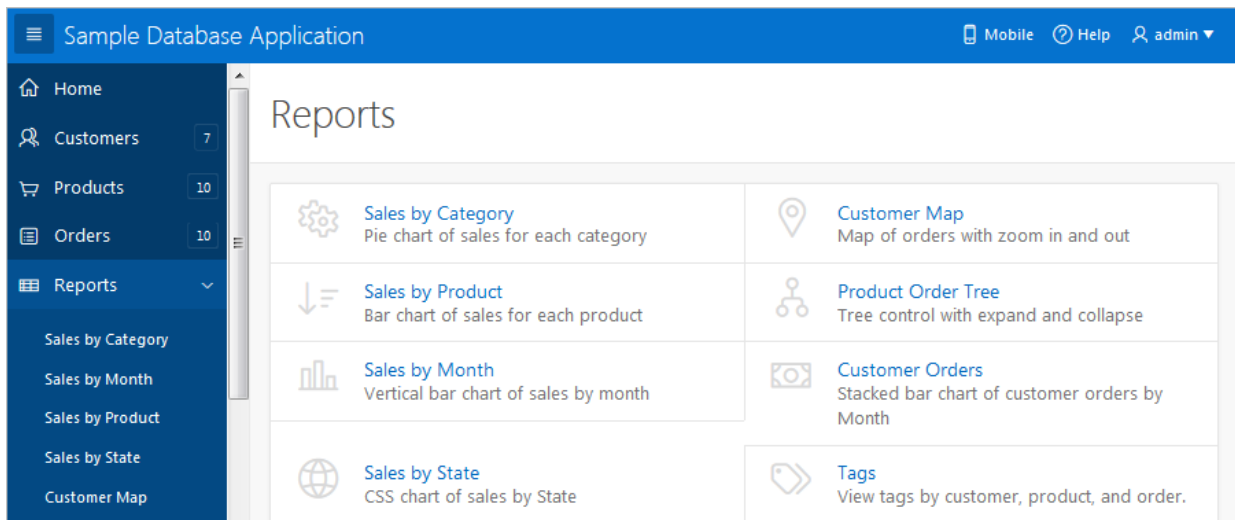
1. Click the Order number.
The Order Details page appears.
2. On the Order Details page, edit the fields and click **Apply Changes**.
3. To delete an order, click **Delete**.

3.3.2.7.2 Placing an Order

To create an order:

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

3.3.2.8 Reports Page



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

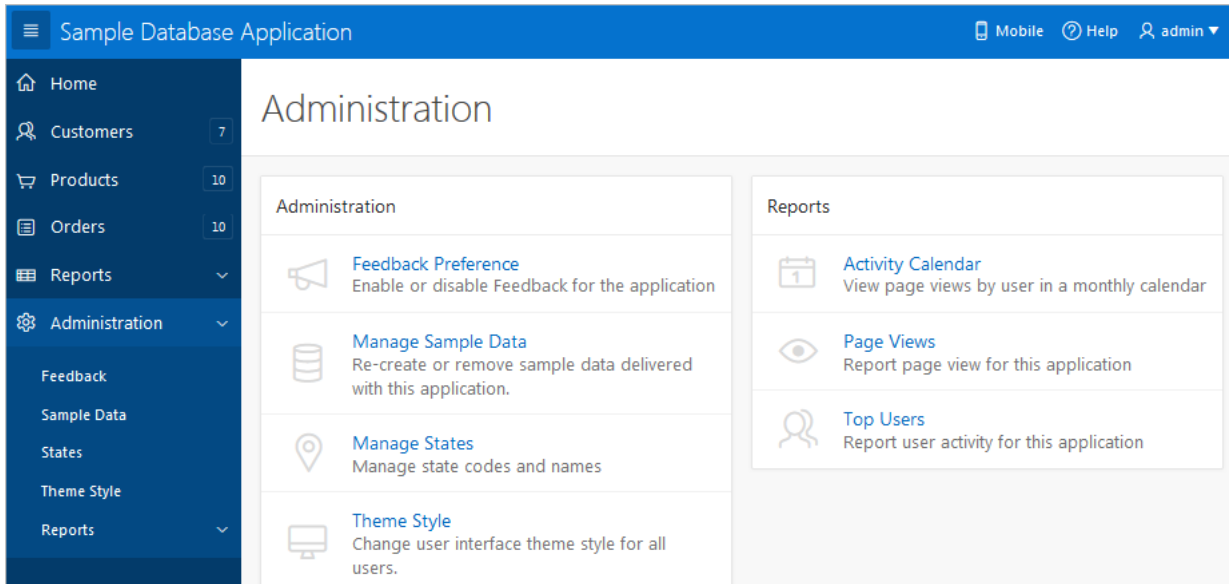
See Also:

- ["Creating Charts \(page 10-23\)"](#)
- ["Creating Maps \(page 10-12\)"](#)
- ["Creating Trees \(page 13-61\)"](#)

3.3.2.9 Administration Page

Use the Administration page to manage sample data, manage state codes and names, and enable user feedback capability. Use the reports on the right side of the page to view user activity, view pages views, and view page views in a monthly calendar.

The Administration contains two regions: Administration and Reports.



The Administration region contains these sections:

- **Feedback Preference** - Enable, disable, and manage feedback for the application.
- **Manage Sample Data** - Remove or re-create the sample data that ships with *Sample Database Application*.
- **Manage States** - Add, edit, or delete states.
- **Theme Styles** - Change the user interface theme style for all users.

The Reports region contains these sections:

- **Activity Calendar** - Displays page views by user in a monthly calendar.
- **Page Views** - Displays a report of page views by user in the application.
- **Top Users** - Displays a report of user activity for this application.

 **See Also:**

- ["Managing Feedback \(page 4-20\)"](#)
- ["Using Theme Styles \(page 11-3\)"](#)

3.3.2.10 Accessing the Mobile Version

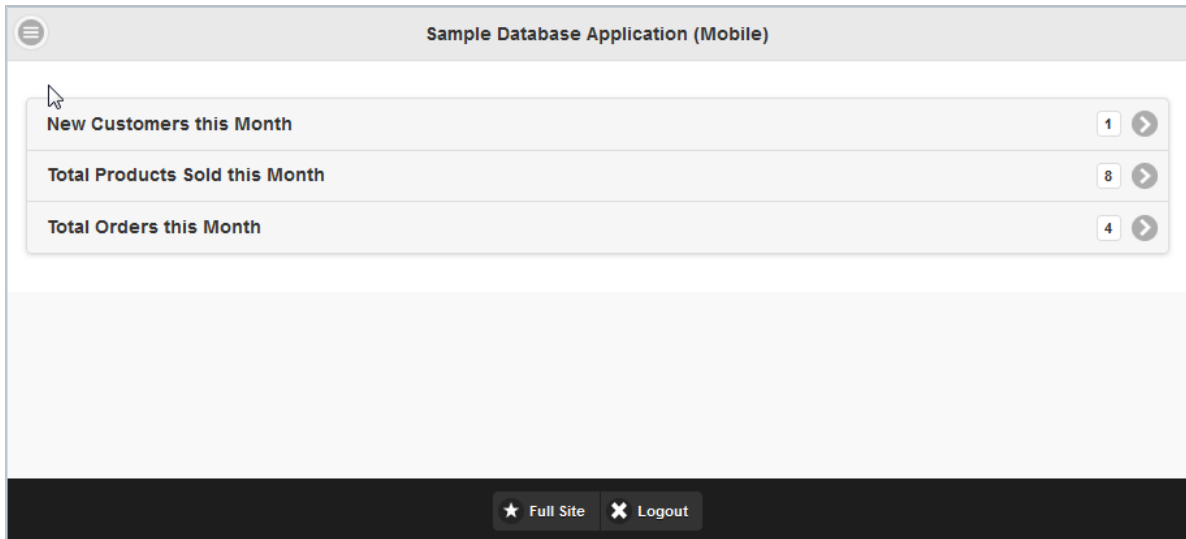
Clicking **Mobile** in the upper right corner of the home page displays *Sample Database Application* in a format compatible with a mobile device. This mobile version replicates the majority of functionality available in the desktop version.

To access the mobile version of *Sample Database Application*:

1. Go to the Packaged Apps page and run *Sample Database Application*.

2. Click **Mobile** in the upper right corner.

The Mobile version of *Sample Database Application* appears.



3. To view a page, select it.
4. To edit a specific customer or product, select it.
A form appears. You can edit the content and click **Apply Changes** . To delete a record , click **Delete**.
5. To return to the home page, click the **Home** icon in the upper left corner.
The home page appears.
6. On the home page:
 - Click **Full Site** to return to the desktop version of *Sample Database Application*.
 - Click **Logout** to return to App Builder.

 **See Also:**

["Running an Installed Packaged Application \(page 3-5\)"](#)

3.3.3 Modifying Sample Database Application

One approach to learning how to build applications in Oracle Application Express is to study *Sample Database Application*.

- [Why Modify a Sample Database Application?](#) (page 3-18)
- [Editing Sample Database Application](#) (page 3-18)



See Also:

"Managing Packaged Applications in a Full Development Environment (page 3-2),"

3.3.3.1 Why Modify a Sample Database Application?

Once you understand the type of functionality available in a packaged 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 packaged applications. If you happen to break something, you can quickly delete the packaged application and install it again.



Tip:

Oracle Application Express includes two types of packaged applications: *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 packaged application makes it ineligible for future upgrades or support by Oracle Support.

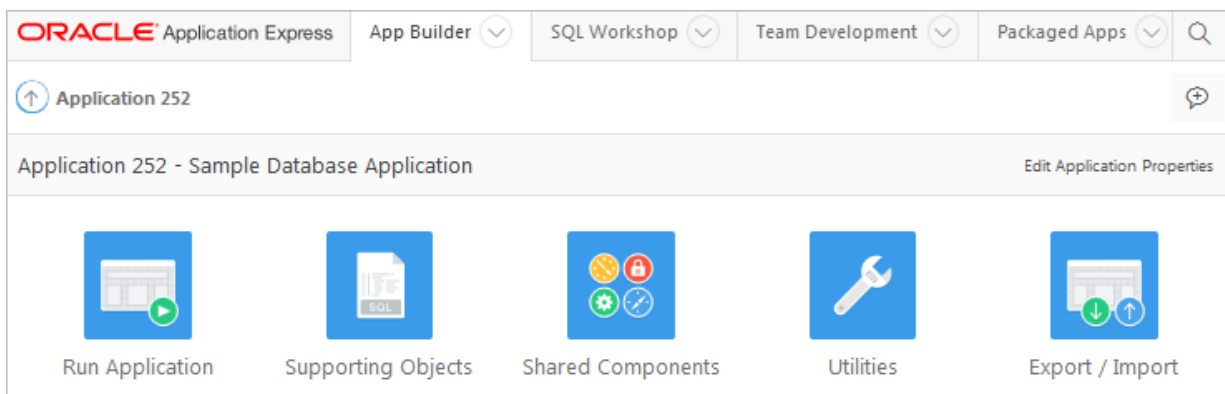
3.3.3.2 Editing Sample Database Application



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. You cannot edit a Modal dialog page from the Runtime Developer toolbar.

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.



The pages that comprise the application appear on the bottom of the 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:

"[Using the Application Home Page](#) (page 1-17)" and "[Runtime Developer Toolbar](#) (page 24-29)"

3.4 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 packaged applications, *AnyCo IT Department* is fully editable.

- [Installing and Running AnyCo IT Department](#) (page 3-19)
- [Understanding Websheet Navigation](#) (page 3-20)
- [Understanding AnyCo IT Department](#) (page 3-21)
- [Modifying Websheet Content](#) (page 3-28)

See Also:

"[Creating Websheet Applications](#) (page 5-1)" and "Using Websheets" in *Oracle Application Express End User Guide*

3.4.1 Installing and Running AnyCo IT Department

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

1. Sign in to Oracle Application Express.

The Workspace home page appears.

2. Click the **Packaged Apps** icon.

The Packaged Apps page appears.

3. Locate and select the *Sample Websheet - AnyCo IT Department*



Tip:

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

The Packaged App Details page appears.

4. Click **Install Packaged App**.
5. Select an Authentication scheme and click **Next**.
6. Click **Install Packaged 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.

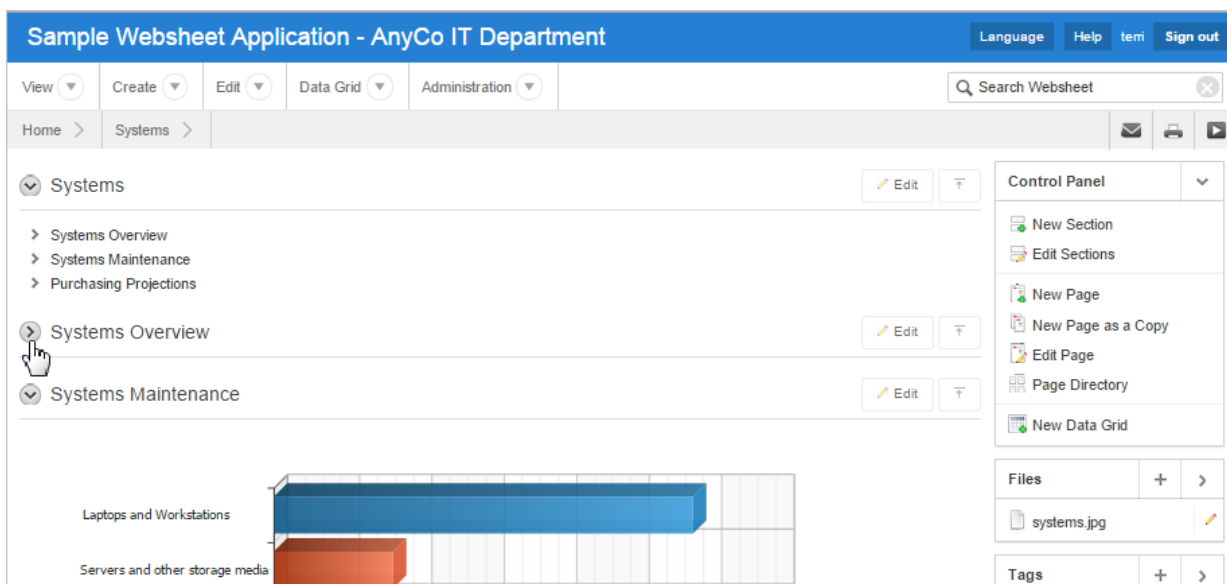


See Also:

["How Your Browser Impacts the Way Applications Run \(page 2-8\)"](#)

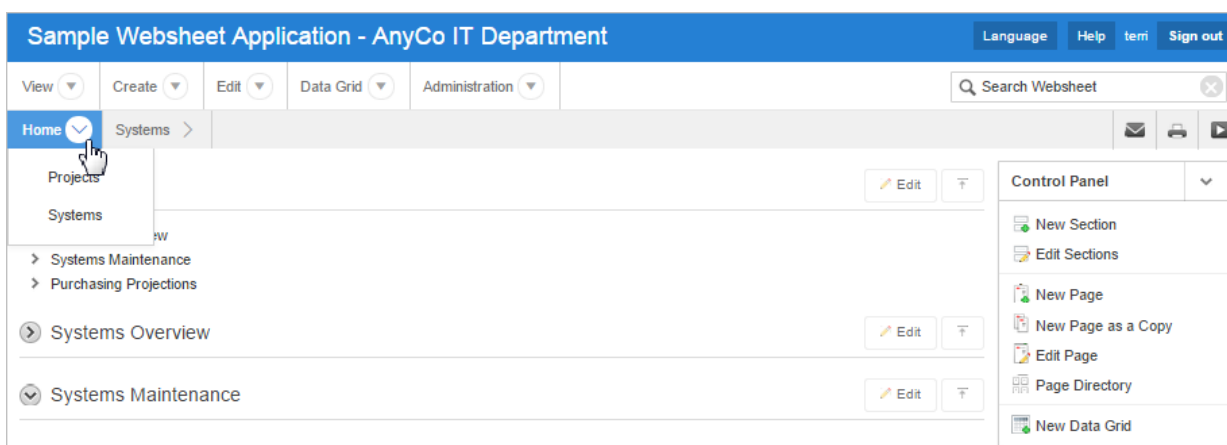
3.4.2 Understanding Websheet Navigation

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.



3.4.3 Understanding AnyCo IT Department

Sample Websheet - AnyCo IT Department demonstrates the following functionality:

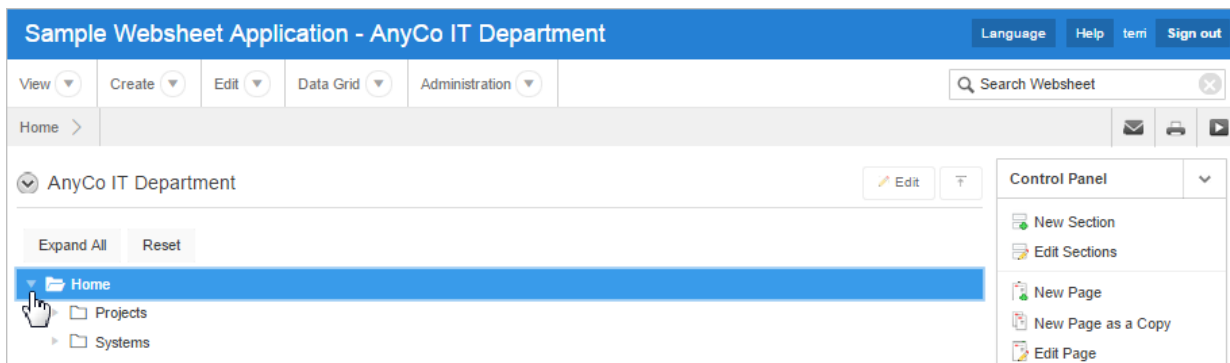
- Examples of ways to share text, data, and images using Websheet pages
- Hierarchical navigation between page sections and pages
- Formatted text that includes easily formatted links
- Display of tabular data by creating Data Grids and displaying the content as reports, interactive reports, or charts.

The following sections describe specific functionality available on each page.

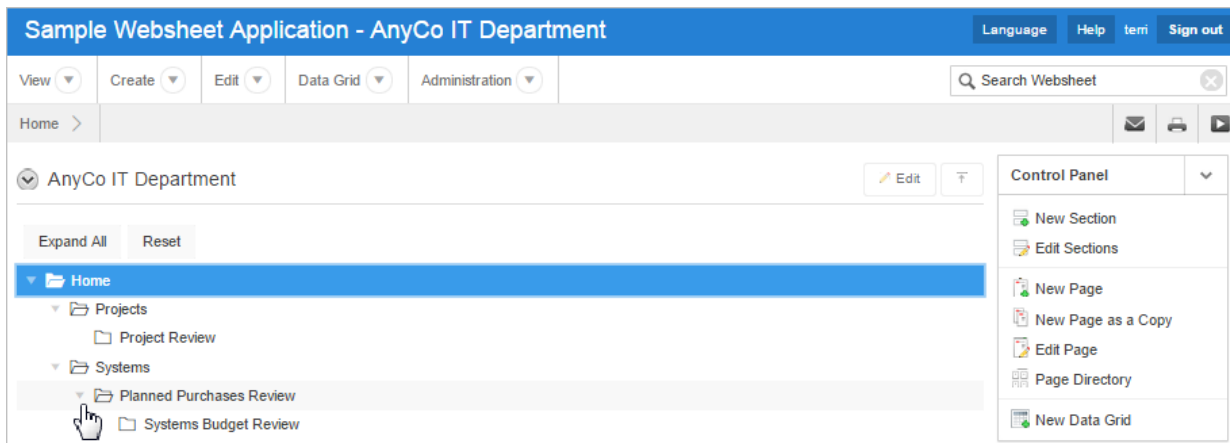
- [Navigating Between Pages and Viewing Page Sections](#) (page 3-22)
- [Home Page](#) (page 3-23)
- [Projects Page](#) (page 3-25)
- [Project Review Page](#) (page 3-25)
- [Systems Page](#) (page 3-26)
- [Planned Purchase Review](#) (page 3-27)
- [Systems Budget Review](#) (page 3-28)

3.4.3.1 Navigating Between Pages and Viewing Page Sections

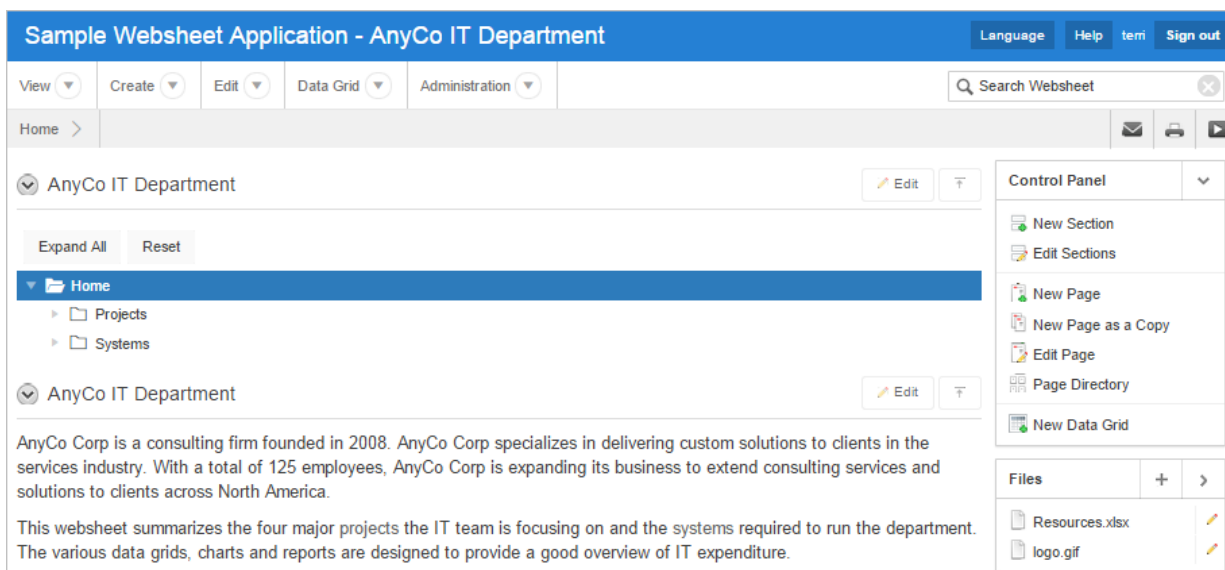
The hierarchical tree displays at the top of the home page enables you to navigate between Websheet pages. Click the arrow to the left of the Home breadcrumb to expand or collapse the tree.



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.



3.4.3.2 Home Page

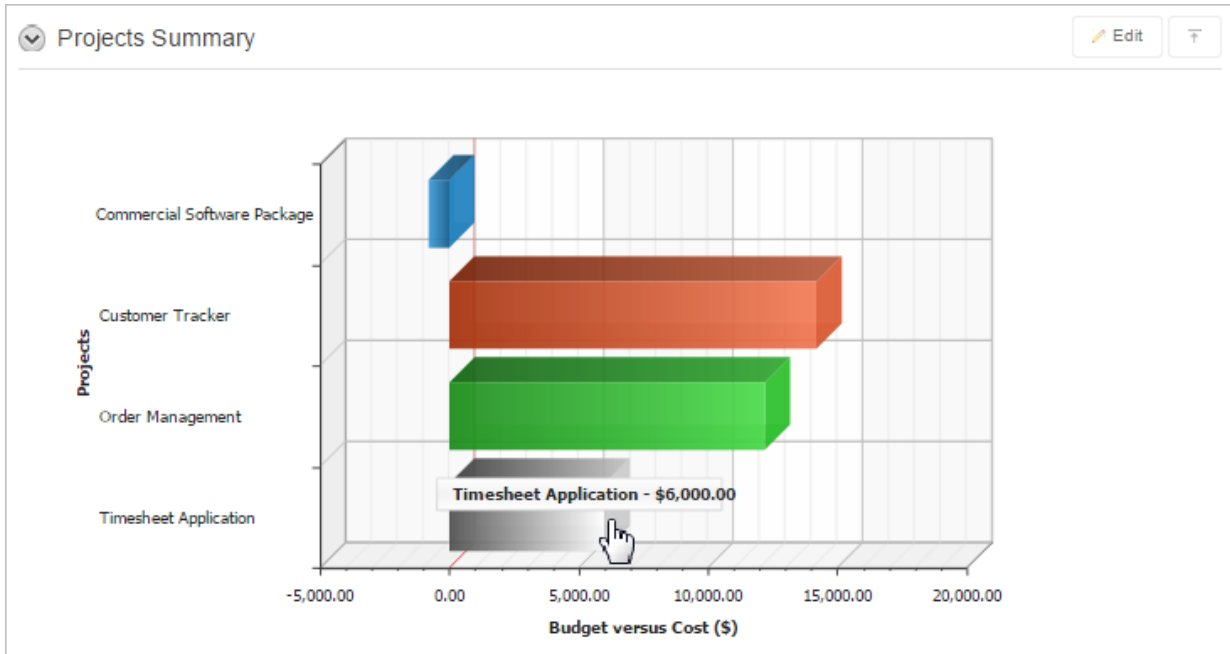


The *Sample Websheet - AnyCo IT Department* home page contains four sections:

- Page navigation tree
- AnyCo IT Department
- Projects Summary
- Systems Summary

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

Projects Summary is a data grid that displays as 3D chart. Passing the cursor over a bar changes the color and displays a dollar amount.



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 ⌵

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

3.4.3.3 Projects Page

Sample Websheet Application - AnyCo IT Department

Language Help term Sign out

View Create Edit Data Grid Administration Search Websheet

Home > Projects >

Projects Breakdown Edit

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

Search Report Search Add Row

Project	Task	Start Date	End Date	Status	Assigned To	Cost	Budget
Timesheet Application	Determine business rules	15-MAR-2010	15-NOV-2010	Open	Pam King	2500	4000
Timesheet Application	Create prototype and trial testing	20-MAR-2010	30-NOV-2010	Open	James Cassidy	6000	10000

Control Panel

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

Files + >

No Files

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.

3.4.3.4 Project Review Page

Sample Websheet Application - AnyCo IT Department

View Create Edit Data Grid Administration

Home > Projects > Project Review

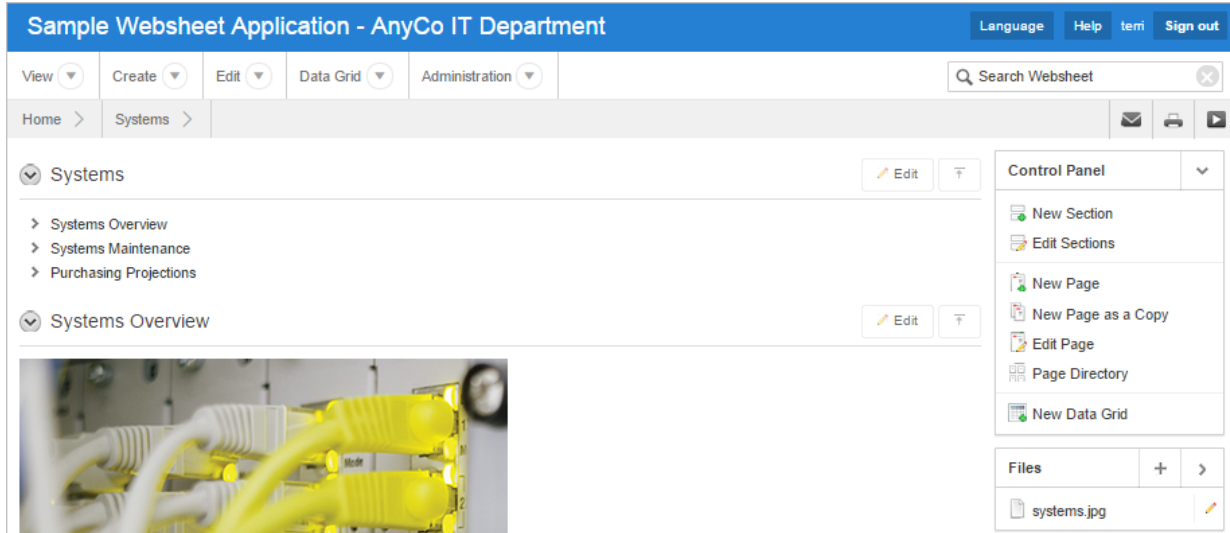
Budget Review Edit

Task Analysis Edit

Task	Value
Develop application to streaml...	~100
Consolidate customer contacts ...	~90
Create prototype and trial tes...	~40
Deliver single customer tracke...	~20
Determine business rules	~10

The Project Review page contains two sections, **Budget Review** and **Task Analysis**. Both sections are data grids that display as different types of 3-D charts.

3.4.3.5 Systems Page



The Systems page contains the following sections:

- Systems (navigation)
- Systems Overview
- Systems Maintenance
- Purchasing Projections

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 as a 3D horizontal chart. Passing the cursor over a bar changes the color and displays a dollar amount.

Purchasing Projections features a report of planned purchases.

3.4.3.6 Planned Purchase Review

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

The 'Total Planned Purchases' section displays a 3D horizontal bar chart with the following data:

Category	Value (Approximate)
Project Management and Account...	High
InitCloud	Low

The Planned Purchase Review page contains two sections:

- Vendor Analysis
- Total Planned Purchases

Vendor Analysis is report of vendors.

Total Planned Purchases is a data grid that displays as 3D horizontal chart. Passing the cursor over a bar changes the color and displays a dollar amount.

3.4.3.7 Systems Budget Review

Home > Systems > Planned Purchases Review > Systems Budget Review

Annual Budget Allocation

The following is the item-wise annual budget allocation:

Search Report [Search]

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](#).

Remaining Budget for the Year

Hardware - \$141,000.00

Cloud - \$6,000.00

The Systems Budget Review page contains two sections:

- Annual Budget Allocation
- Remaining Budget for the Year

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 as 3D pie chart. Passing the cursor over a pie section displays the dollar amount. Selecting a section, removes it slightly from the pie.

3.4.4 Modifying Websheet Content

Once you understand the type of functionality available in *AnyCo IT Department*, the next step is to learn more about the construction of each page.

- [Why Modify Sample Websheet Application?](#) (page 3-29)
- [Updating a Websheet Section](#) (page 3-29)

3.4.4.1 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. See "[Managing Packaged Applications in a Full Development Environment](#) (page 3-2)."

 **Tip:**

Oracle Application Express includes two types of packaged applications: *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 packaged application makes it ineligible for future upgrades or support by Oracle Support.

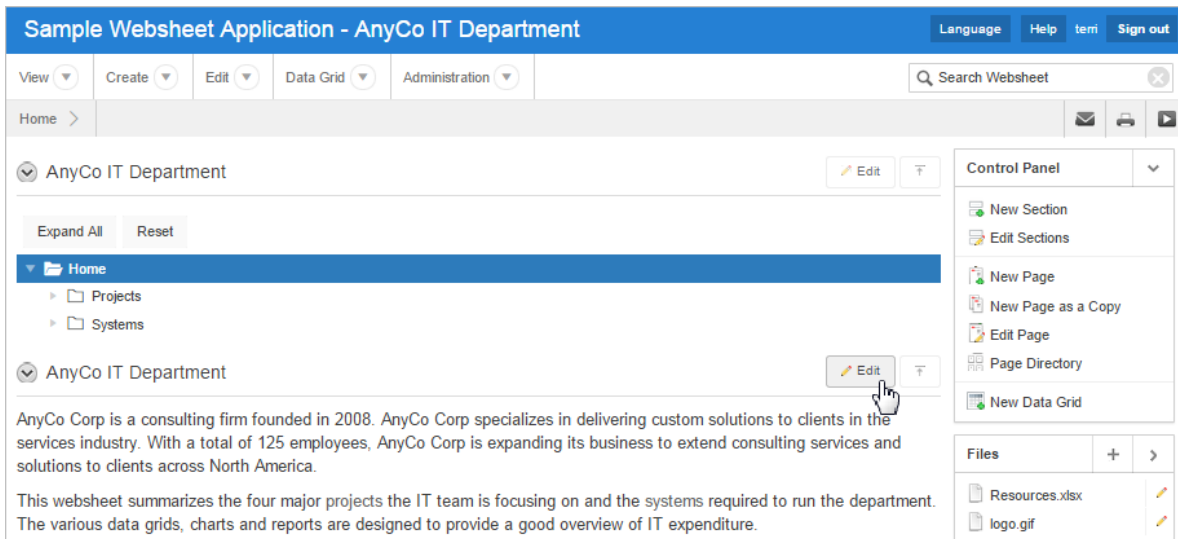
 **See Also:**

"[Creating Websheet Applications](#) (page 5-1)"

3.4.4.2 Updating a Websheet Section

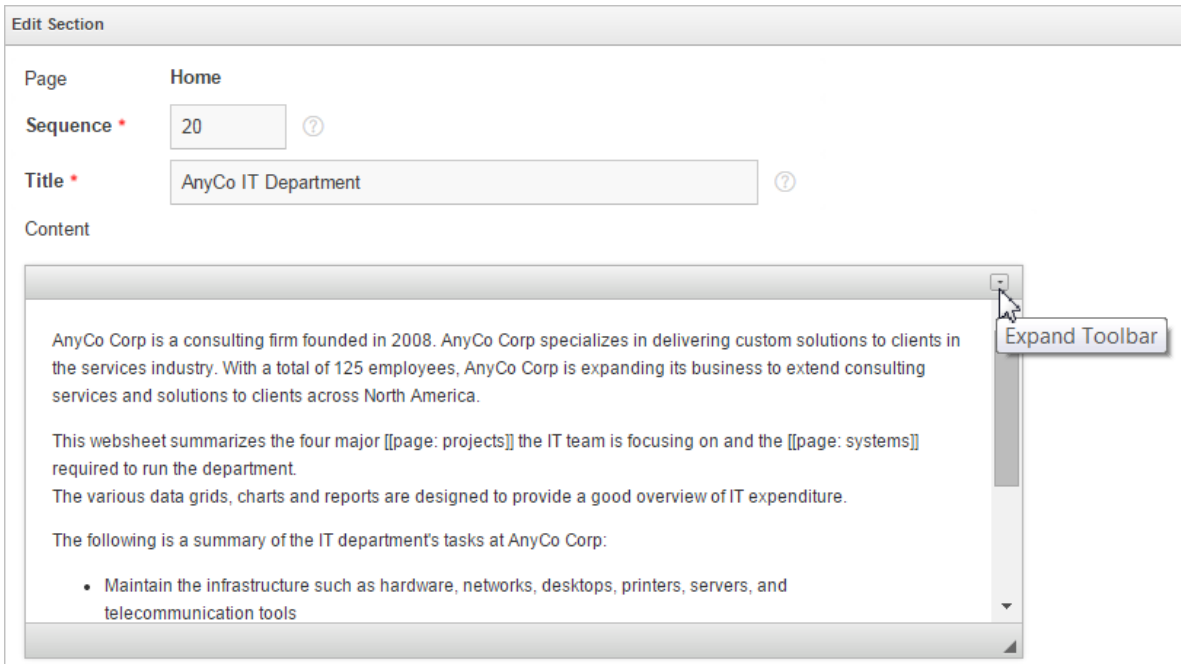
To update a Websheet section:

1. Run the Websheet.
2. Click the Edit icon that displays in the upper right of the section.



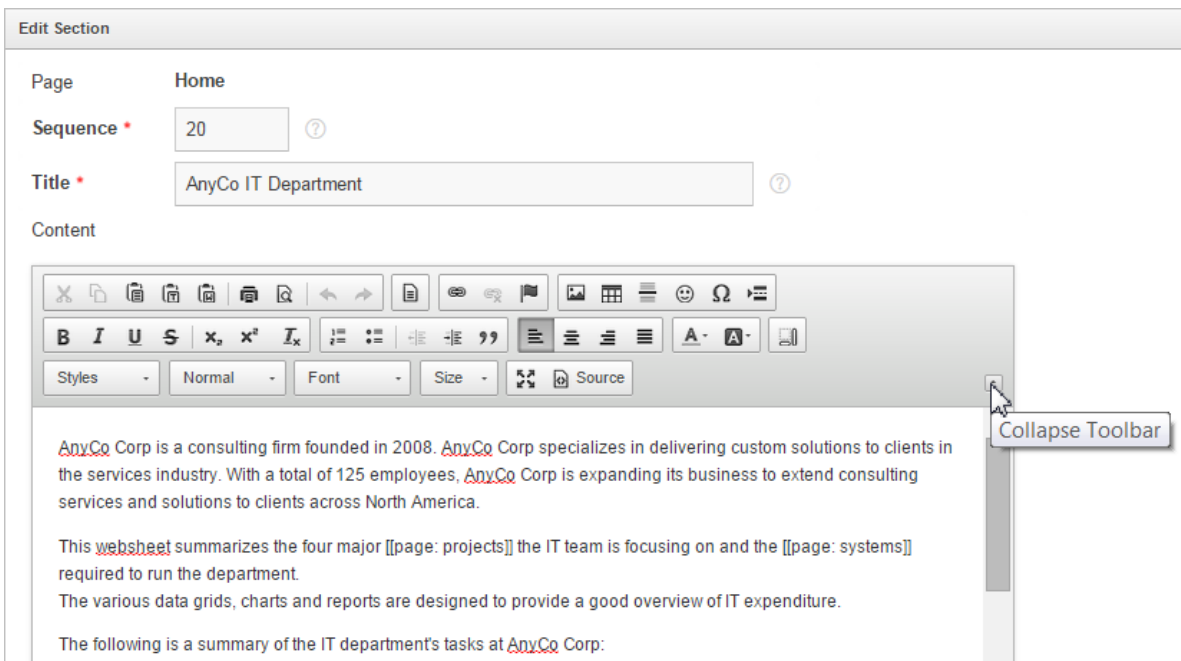
The Edit Section page appears. How you edit a specific section depends upon the section type.

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.



6. To save your changes, click **Apply Changes**.



Note:

"[About Packaged Applications](#) (page 3-1)"

3.5 Utilizing Packaged App Administration

Administrators use Packaged App Administration to manage and monitor packaged apps in their workspace.

- [Making a Service Request](#) (page 3-31)
- [Setting Workspace Preferences](#) (page 3-32)
- [Managing Users](#) (page 3-33)
- [Managing Saved Interactive Reports](#) (page 3-33)
- [Managing Interactive Report Subscriptions](#) (page 3-34)
- [Viewing Activity Reports](#) (page 3-34)
- [Viewing Page Views](#) (page 3-34)
- [Viewing Top Users](#) (page 3-35)



See Also:

"Workspace and Application Administration" in *Oracle Application Express Administration Guide*

3.5.1 Making a Service Request



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 **Packaged Apps**.
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 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.

3.5.2 Setting Workspace Preferences

 **Tip:**

Only users with administrator rights can access this functionality.

Administrators can configure account login controls on the Set Workspace Preferences page.

To set workspace preferences:

1. Sign in to Oracle Application Express.
2. Click **Packaged Apps**.
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**.

3.5.3 Managing Users

 **Tip:**

Only users with administrator rights can access this functionality. This option is only available in a Packaged Apps Only workspace, but not in full workspace.

Click **Manage Users** to create new user accounts, manage existing user accounts, and change user passwords.

To manage users in a Packaged Apps Only workspace:

1. Sign in to the Packaged Apps Only workspace.
2. Click **Packaged App Administration**.
3. Click **Manage Users**.

Managing users works the same in a Packaged Apps Only workspace as it does at the workspace-level.

 **See Also:**

"Managing Users in a Workspace" in *Oracle Application Express Administration Guide*

3.5.4 Managing Saved Interactive Reports

Click **Saved Reports** to view and manage interactive reports created by users of packaged applications.

To manage saved interactive reports:

1. Sign in to Oracle Application Express.
2. Click **Packaged Apps**.
3. Click **Administration**.
4. Click **Saved Reports**.

Managing saved interactive reports works the same in a Packaged Apps Only workspace as it does at the workspace-level.

 **See Also:**

"Viewing and Deleting Saved Interactive Reports" in *Oracle Application Express Administration Guide*

3.5.5 Managing Interactive Report Subscriptions

Click **Subscriptions** to view and manage interactive report subscriptions created by users of packaged applications.

To manage interactive report subscriptions:

1. Sign in to Oracle Application Express.
2. Click **Packaged Apps**.
3. Click **Administration**.
4. Click **Subscriptions**.

Managing interactive report subscriptions works the same in a Packaged Apps Only workspace as it does at the workspace-level.



See Also:

"Managing Interactive Report Subscriptions" in *Oracle Application Express Administration Guide*

3.5.6 Viewing Activity Reports

Click **Application Activity** to view a monthly calendar of application activity reporting distinct users and total page views by day.

To view Application Activity report:

1. Sign in to Oracle Application Express.
2. Click **Packaged Apps**.
3. Click **Administration**.
4. Click **Packaged App Activity**.

A monthly calendar of application activity appears.

3.5.7 Viewing Page Views

Click **Page Views** to view an interactive report of each page view detailing the user, elapsed time, and page.

To view the Page View report:

1. Sign in to Oracle Application Express.
2. Click **Packaged Apps**.
3. Click **Administration**.
4. Click **Page Views**.

3.5.8 Viewing Top Users

Click **Top Users** to view a report of page views aggregated by user.

To view the Top Users report:

1. Sign in to Oracle Application Express.
2. Click **Packaged Apps**.
3. Click **Administration**.
4. Click **Top Users**.

3.6 Using a Packaged Apps Only Workspace

A **Packaged Apps Only** workspace enables developers to share packaged applications with other developers and administrators.



Note:

This information does not apply to Oracle Database Cloud Service (Database Schema).

- [What Is a Packaged Apps Only Workspace?](#) (page 3-35)
- [Requesting a Packaged Apps Only Workspace](#) (page 3-35)
- [Signing In To a Packaged Apps Only Workspace](#) (page 3-36)
- [Using the Packaged App Page](#) (page 3-37)

3.6.1 What Is a Packaged Apps Only Workspace?

A Packaged Apps Only workspace enables developers to share packaged applications with other developers and administrators. It does not include any access to development tools such as App Builder or SQL Workshop. A Packaged Apps Only workspace includes the Packaged Apps page and the Packaged App Administration page.



Tip:

End users cannot access a Packaged Apps Only workspace. End users can only run existing database or Websheet applications.

3.6.2 Requesting a Packaged Apps Only Workspace

To request a Packaged 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 **Packaged 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*.

3.6.3 Signing In To a Packaged Apps Only Workspace

 **Tip:**

Sample applications are not available in a Packaged App Only workspace. Otherwise, the Packaged Apps page in a Packaged Apps Only workspace works the same as in a full development environment. See "[Managing Packaged Applications in a Full Development Environment](#) (page 3-2)."

To sign in to a Packaged Apps Only workspace:

1. Sign in to the Packaged Apps Only workspace.
2. Click the **Packaged Apps** icon.

The Packaged Apps page appears.

3. Click **Packaged App Administration** to access the Packaged Application Administration page.

 **See Also:**

"[Utilizing Packaged App Administration](#) (page 3-31)."

3.6.4 Using the Packaged App Page

 **Tip:**

Sample applications are not available in a Packaged App Only workspace. Otherwise, the Packaged Apps page in a Packaged Apps Only workspace works the same as in a full development environment. See "[Managing Packaged Applications in a Full Development Environment](#) (page 3-2)."

To access the Packaged App page:

1. Sign in to the Packaged Apps Only workspace.
2. Click the **Packaged Apps** icon.
The Packaged Apps page appears.
3. To search for an application, enter search terms in Search field.
4. Select an application to view.

The Packaged Application Details page appears. Use this page to view information about a packaged application, install an application, remove an application, or change the authentication of a previously installed packaged 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 provide real-time feedback which then can be categorized into To Dos, bugs, or features.

Topics:

- [Accessing Team Development](#) (page 4-1)
- [Team Development Home Page](#) (page 4-2)
- [About Utilizing the Tags Summary](#) (page 4-3)
- [Tracking Features](#) (page 4-3)
- [Tracking Milestones](#) (page 4-10)
- [Tracking To Dos](#) (page 4-13)
- [Managing Bugs](#) (page 4-16)
- [Managing Feedback](#) (page 4-20)
- [Managing Team Development Utilities](#) (page 4-24)

See Also:

"Managing Team Development" in *Oracle Application Express Administration Guide*

4.1 Accessing Team Development

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.

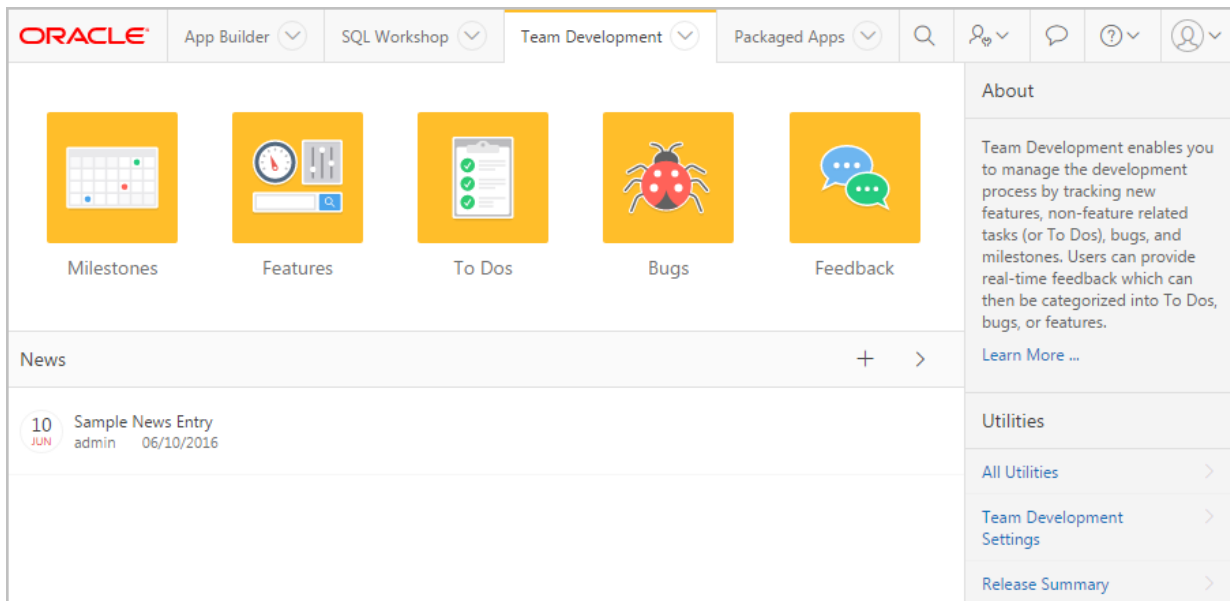
See Also:

"[Team Development Home Page](#) (page 4-2)"

4.2 Team Development Home Page

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 \(page 4-28\)"](#)

4.3 About Utilizing the Tags Summary

The Tags summary on the Team Development home page 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 Tag Cloud summary to better manage Team Development components.

To populate this summary, edit the Tags field for each Team Development component (that is, feature, To Do task, bug, and feedback entry).

4.4 Tracking Features

Use the Features page to track features from initial concept through implementation. You can organize features by release, assignee, tags, or associated milestones.

- [Viewing the Features Report \(page 4-4\)](#)

- [Creating a Feature](#) (page 4-4)
- [About Creating Parent Features](#) (page 4-5)
- [Updating a Feature](#) (page 4-5)
- [How to Attach Files to a Feature](#) (page 4-5)
- [Viewing Other Feature Reports](#) (page 4-7)

4.4.1 Viewing the Features Report

To view features:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Features** icon.

The Features Dashboard appears. 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. To customize the appearance of the page, use the Search bar at the top of the page.



See Also:

["Customizing Interactive Reports in a Running Application \(page 8-92\)"](#) and ["Updating a Feature \(page 4-5\)"](#)

4.4.2 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 appears.

3. Click **Create Feature**.
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 Feature**.

 **See Also:**

["Editing Focus Areas \(page 4-27\)"](#) and ["Viewing Field-Level Help \(page 1-27\)"](#)

4.4.3 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 \(page 4-7\)"](#)

4.4.4 Updating a Feature

To update 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 edit a feature, click the feature name.
5. Edit the appropriate fields.
6. Click **Apply Changes**.

 **See Also:**

["Editing Focus Areas \(page 4-27\)"](#)

4.4.5 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 \(page 4-6\)](#)
- [Viewing, Editing, and Removing Attached Files \(page 4-6\)](#)



See Also:

"[Viewing File Attachments](#) (page 4-27)" and "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*.

4.4.5.1 Attaching a File to a Feature



Tip:

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. Navigate to the Feature.
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 Feature page.



Note:

"[Updating a Feature](#) (page 4-5)"

4.4.5.2 Viewing, Editing, and Removing Attached Files

To view, edit, or remove attached files:

1. Navigate to the Feature page. See "[Updating a Feature](#) (page 4-5)."
2. Scroll down to Files.
3. Click **View and Manage Files**.

The Files page appears.
4. To view a file, click the name.
5. To edit the file description, click the **Edit** icon.
6. To delete an attached file, click the red **Delete** icon.

4.4.6 Viewing Other Feature Reports

This section describes feature reports available in Team Development.

- [Viewing the Features Tree](#) (page 4-7)
- [Viewing the Features Calendar](#) (page 4-8)
- [Viewing The Feature History Report](#) (page 4-8)
- [Viewing the Feature Progress Log](#) (page 4-8)
- [Viewing the Feature Focus Areas Chart](#) (page 4-9)
- [Viewing the Approval Statuses](#) (page 4-9)
- [Viewing the Feature Owners Chart](#) (page 4-9)
- [Feature Utilities](#) (page 4-10)



See Also:

["Accessing Team Development Utilities](#) (page 4-24)"

4.4.6.1 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.

To select a new release, make a selection from the Release list and click **Set**. 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**. To edit a feature, simply click it.

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.

Click the **Expand All** and **Collapse All** buttons to expand or collapse all grouping. To view or edit a feature, click the feature name. To create a feature, click **Create Feature** and follow the online instructions.

4.4.6.2 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. Click the **Previous** and **Next** buttons to view other months. To view feature milestones for the current day, click **Today**. To view a specific feature, click the feature name. To create a feature, click **Create Feature** and follow the online instructions.

4.4.6.3 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. To customize the report, use the Search bar at the top of the page.

4. To view a specific feature, click the feature name.



See Also:

["Customizing Interactive Reports in a Running Application \(page 8-92\)"](#) and ["About the Actions Menu \(page 8-94\)"](#)

4.4.6.4 Viewing the Feature Progress Log

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. To customize the report, use the Search bar at the top of the page.

4. To edit the log, click the **Edit** icon.

 **See Also:**

"[Customizing Interactive Reports in a Running Application](#) (page 8-92)" and "[About the Actions Menu](#) (page 8-94)"

4.4.6.5 Viewing the Feature Focus Areas Chart

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. To alter the chart, select another release or another assignee and click **Set**. To return to the default chart, click **Reset**.

 **See Also:**

"[Editing Focus Areas](#) (page 4-27)"

4.4.6.6 Viewing the Approval Statuses

To view the 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**.

4.4.6.7 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. To alter the chart, select another release, Minimum Status, or Maximum Status, and click **Set**. To return to the default chart, click **Reset**.

4.4.6.8 Feature Utilities

Use the Feature Utilities - Choose Action page to manage milestones for multiple features. Select one of the following and follow the on-screen instructions:

- Assign milestones to features (only those with no milestone currently assigned).
- Set feature due date to milestone date (only affects features).
- Change milestone for overdue features.
- Push due date of open features.



See Also:

["Accessing Team Development Utilities \(page 4-24\)"](#)

4.5 Tracking Milestones

Use the Milestones page to manage important milestones. Milestones track events. You can associate milestones with features, bugs, and To Dos.

- [Viewing the Milestones Report \(page 4-10\)](#)
- [Creating a Milestone \(page 4-11\)](#)
- [Updating a Milestone \(page 4-11\)](#)
- [Viewing Other Milestone Reports \(page 4-11\)](#)

4.5.1 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 (page 8-92)" and
"Updating a Milestone (page 4-11)"

4.5.2 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**.

4.5.3 Updating 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 **Milestones** tab.
The Milestones Report page appears.
4. Select a milestone.
5. Edit the appropriate fields.
To learn more about an attribute, see field-level Help.
6. Click **Apply Changes**.

4.5.4 Viewing Other Milestone Reports

This section describes milestone reports available in Team Development.

- [Viewing the Milestone Calendar](#) (page 4-12)
- [Viewing the Milestones By Owner Report](#) (page 4-12)
- [Viewing the Features by Milestone Report](#) (page 4-12)



See Also:

["Accessing Team Development Utilities \(page 4-24\)"](#)

4.5.4.1 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.

To filter by release, make a selection from the Release list and click **Set**. Click the **Previous** and **Next** buttons to view other months. To view milestones for the current day, click **Today**. To view a specific milestone, click the hyperlink. To create a milestone, click **Create Milestone** and following the online instructions.

4.5.4.2 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**.

4.5.4.3 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. 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 (page 8-92)" and "About the Actions Menu (page 8-94)"

4.6 Tracking To Dos

To Dos track work assignments as opposed to functionality. They can be associated with features and milestones to define the tasks needed for completion. To Dos can be organized by release, assigned to developers, and tagged.

- [Viewing To Dos](#) (page 4-13)
- [Creating a To Do](#) (page 4-13)
- [Updating a To Do](#) (page 4-14)
- [Managing Files Attached to a To Do](#) (page 4-14)
- [Viewing Other To Do Reports](#) (page 4-15)

4.6.1 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.

The To Dos page displays an interactive report of current tasks. You can customize the appearance of page, using the Search bar at the top of the page. To view your personal To Dos, click the **My To Dos** button at the top of the page.

 **See Also:**

"Customizing Interactive Reports in a Running Application (page 8-92)" and "Updating a To Do (page 4-14)"

4.6.2 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**.



See Also:

["Viewing Field-Level Help \(page 1-27\)"](#)

4.6.3 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.
4. Select a To Do.
5. Edit the appropriate fields.
6. Click **Apply Changes**.

4.6.4 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 \(page 4-15\)](#)
- [Viewing, Editing, and Removing Attached Files \(page 4-15\)](#)



See Also:

" [Viewing File Attachments \(page 4-27\)](#)" and "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*.

4.6.4.1 Attaching a File to a To Do

 **Tip:**

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.

4.6.4.2 Viewing, Editing, and Removing Attached Files

To manage attached files:

1. Navigate to the To Do page. See "[Updating a To Do](#) (page 4-14)."
2. Scroll down to Files.
3. Click **View and Manage Files**.
The Files page appears.
4. To view a file, click the name.
5. To edit the file description, click the **Edit** icon.
6. To delete an attached file, click the **Delete** icon.

4.6.5 Viewing Other To Do Reports

This section describes To Do reports available in Team Development.

- [Viewing the To Do Calendar](#) (page 4-16)
- [Viewing the To Do Progress Log](#) (page 4-16)

4.6.5.1 Viewing the To Do Calendar

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 **Calendar** tab.

The To Dos Calendar appears.

The Calendar page displays To Dos by due date or date completed in a calendar format. To filter the display by completion status or release, make a selection from the Show and Release select lists and click **Set**. Click the **Previous** and **Next** buttons to view other months. To view To Dos for the current day, click **Today**. To view a specific To Do, click the To Do name. To create a To Do, click **Create To Do** and follow the online instructions.

4.6.5.2 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. To edit the log, click the **Edit** icon.

See Also:

["Customizing Interactive Reports in a Running Application \(page 8-92\)"](#) and ["About the Actions Menu \(page 8-94\)"](#)

4.7 Managing Bugs

Bugs track software defects. Bugs can be assigned, associated with milestones, and tracked by due date, status, and other attributes.

- [Viewing and Editing Bugs \(page 4-17\)](#)
- [Creating a Bug \(page 4-17\)](#)
- [Editing a Bug \(page 4-17\)](#)
- [Managing Files Attached to a Bug \(page 4-18\)](#)
- [Viewing Other Bug Reports \(page 4-19\)](#)

4.7.1 Viewing and Editing 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 \(page 8-92\)"](#) and ["Editing a Bug \(page 4-17\)"](#)

4.7.2 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.
The value you enter is added.
6. Click **Create Bug**.



See Also:

["Viewing Field-Level Help \(page 1-27\)"](#)

4.7.3 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**.



Note:

["Viewing Field-Level Help \(page 1-27\)"](#)

4.7.4 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 \(page 4-18\)](#)
- [Viewing, Editing, and Removing Attached Files \(page 4-19\)](#)



See Also:

"[Viewing File Attachments \(page 4-27\)](#)" and "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*.

4.7.4.1 Attaching a File to a Bug



Tip:

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. To edit a bug, 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.

4.7.4.2 Viewing, Editing, and Removing Attached Files

To manage attached files:

1. Navigate to the Bug page. See "[Editing a Bug](#) (page 4-17)."
2. Scroll down to Files.
3. Click **View and Manage Files**.

The Files page appears.
4. To view a file, click the name.
5. To edit the file description, click the **Edit** icon.
6. To delete an attached file, click the **Delete** icon.

4.7.5 Viewing Other Bug Reports

This section describes bug reports available in Team Development.

- [Bugs Dashboard](#) (page 4-19)
- [Viewing the Bug Calendar](#) (page 4-20)
- [Viewing the Bugs By Developer](#) (page 4-20)
- [Viewing the Bugs By Day](#) (page 4-20)

4.7.5.1 Bugs Dashboard

The Bugs Dashboard displays an overview of known software defects. The Dashboard page contains the following regions:

- Percent Closed
- Assigned Developer
- Severity
- Status

To edit the dashboard, edit the Show and Release lists at the top of the page and click **Set**. To reset the dashboard to the default display, click the **Reset** button.

4.7.5.2 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.

Use the Show list on the Calendar page to display open bugs with an estimated close date, open bugs with no estimated close date, or closed bugs. Click the **Previous** and **Next** buttons to view other months. To view a specific bug, click the bug description. To create a new bug, click **Create Bug**.

4.7.5.3 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 By Developer page displays bugs by developer. To view bugs associated with a specific assignee, click the appropriate section of the chart. A report appears displaying bugs by developer.

To alter the chart, select another release and click **Set**. To return to the default chart, click **Reset**. To create a new bug, click **Create Bug**.

4.7.5.4 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 By Day page displays number of bugs opened and closed by day. To alter the chart, select another release and click **Set**. To return to the default bar chart, click **Reset**. To create a new bug, click **Create Bug**.

4.8 Managing Feedback

Feedback is the process of gathering real-time comments, enhancement requests, and bugs from your application users. To gather feedback for an application, you add a Feedback page.

- [Viewing Feedback](#) (page 4-21)
- [Adding a Feedback Page to an Application](#) (page 4-21)
- [Submitting Feedback](#) (page 4-22)
- [Reviewing Feedback](#) (page 4-22)
- [Viewing Other Feedback Reports](#) (page 4-23)

4.8.1 Viewing Feedback

To view feedback:

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.

The Feedback Report page displays details about feedback entries. You can further customize the appearance of the page using the Search bar at the top of the page.



See Also:

["Customizing Interactive Reports in a Running Application \(page 8-92\)"](#)

4.8.2 Adding a Feedback Page to an Application

A feedback page enables you to collect feedback from application users.

To add a feedback page to an application:

1. On the Workspace home page, click **App Builder**.
2. Select the application for which you want to add feedback.
The Application home page appears.
3. Click the **Create Page** button.
4. For Page Type, select **Feedback Page**.
5. For Define:
 - a. Page Number - Accept the default or provide an integer value that identifies a page within an application.
 - b. Page Name - Specify a text name for this page.
 - c. Page Mode - Identify the page mode. To learn more about an option, see field-level Help.
 - d. Popup Page Template - Select the page template for the popup page to be used to gather the feedback.
 - e. Region Template - Select a region template for the region in which the feedback is gathered.

- f. Label Template - Select a label template to be used for the items on the feedback page.
- g. Extra Attributes - Specify how many extra attributes should be included on the feedback page.

 **Tip:**

If you specify extra attributes, then you need to edit them to update their labels, display, and help text. Then, you need to update the process called to save the feedback to pass the proper context for each extra attribute so it can be displayed when reviewing the feedback.

- h. Create Navigation Bar - Specify whether to create a navigation bar entry within your current application to invoke your new feedback page.
- i. Entry Label - Enter the name of the new navigation bar entry used to invoke the new feedback page.
- j. Enable Feedback - This attribute only displays if feedback support for the current application is not enabled. To enable feedback support, select **Yes**.

 **Tip:**

To learn more about the application-level attribute, Allow Feedback see "[Editing Application Attributes](#) (page 6-16)"

- 6. Click **Create**.

4.8.3 Submitting Feedback

To submit feedback:

1. Run the application for which you want to submit feedback.
2. Click the **Feedback** link.
A Feedback window appears. The application and page display at the top of the window. This information is made available to the developers.
3. In Feedback, enter a comment.
4. From Feedback Type, select **General Comment**, **Enhancement Request**, or **Bug**.
5. Click **Submit Feedback**.

4.8.4 Reviewing Feedback

To review feedback:

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 and then review the feedback.
5. Under Disposition:
 - a. Status - Select a new status.
 - b. Tags are keywords that further describe this feedback. Separate multiple tags with commas.
 - c. Developer Comment - Enter additional comments. Note Developer Comments are not publicly viewable.
 - d. Public Response - Enter a public response. This response is publicly viewable.
6. Determine how to log the feedback. To the right of Disposition, click 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 **Log as Feature**.To learn more about an option, see field-level Help.
7. To delete a feedback entry, click **Delete**.
8. To submit your changes, click **Apply Changes**.

4.8.5 Viewing Other Feedback Reports

This section describes feedback reports available in Team Development.

- [Viewing the Feedback Calendar](#) (page 4-23)
- [Viewing Feedback By Filing User](#) (page 4-23)

4.8.5.1 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 **Feedback Calendar** tab.

The Calendar page displays the number of feedback comments provided by application by day. Click the **Previous** and **Next** buttons to view other months.

4.8.5.2 Viewing Feedback By Filing User

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 **By Filing User** tab.

The By Filing User page displays a chart of open feedback by filing user. Click a section of the chart to view a more detailed report.

4.9 Managing Team Development Utilities

The Utilities region displays on the right side of the Team Development home page. Use these links 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.

- [Accessing Team Development Utilities](#) (page 4-24)
- [Configuring Team Development Settings](#) (page 4-26)
- [Viewing a Release Summary](#) (page 4-26)
- [Managing Feature Utilities](#) (page 4-27)
- [Editing Focus Areas](#) (page 4-27)
- [Viewing File Attachments](#) (page 4-27)
- [Purging Data](#) (page 4-28)
- [Managing News Entries](#) (page 4-28)
- [Managing Links](#) (page 4-29)



See Also:

"Managing Team Development" in *Oracle Application Express Administration Guide*

4.9.1 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.
- **Enable Files** - Links to the administration page that enables you to add support for file upload.
- **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.

 **Tip:**

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 \(page 4-26\)"](#)
- "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*

4.9.2 Configuring Team Development Settings

Team Development Settings enable you specify how To Dos and bugs display, default values for release, owner, and priority, and what tracking 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.
4. Click **Apply Changes**.



Tip:

Click **Reset to Default** to return the settings to the default values.

4.9.3 Viewing a Release Summary

The Release Summary report enables you to organize features, To Dos, and bugs.

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 to restrict the view. To filter the view, select a developer or release and click **Set**.
4. To email the report:
 - a. Click **Email**.
The Email Report page appears.
 - b. Fill in the appropriate fields. Mandatory fields are marked with a red asterisk (*). To learn more about an attribute, see field-level Help.
5. Click **Email**.



Tip:

Click **Reset** to return the report to the default view.

4.9.4 Managing Feature Utilities

Use the Feature Utilities page 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 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)**
 - **Change milestone for overdue features**
 - **Push due date of open features**
4. Click **Next**.
5. Follow the on-screen instructions.

4.9.5 Editing Focus Areas

When you create or update a feature, you can assign it to a Focus Area.

To edit 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.
5. Click **Apply Changes**.

4.9.6 Viewing File Attachments

If enabled at the workspace-level, you can attach files to a feature, To Do, or bug.

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 view 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 view a file, click the name.



See Also:

"Enabling Support for File Attachments in Team Development" in *Oracle Application Express Administration Guide*.

4.9.7 Purging Data

Purging data removes all the data associated with a specific type of data (that is, features, milestones, To Dos, bugs, or feedback).

To purge data:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click the **Purge Data**.
3. Under Purge Data, select the types of data to purge and click **Purge**.

4.9.8 Managing News Entries

Use the News region to communicate with other users in the current workspace. You can add or view news entries posted by other workspace users. News displays on the Application Express home page, the Team Development home page, and News page.

To manage new entries:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Manage News**.
The News page appears.
3. Use the Search bar to filter the display.
4. To add a news item:
 - a. Click **Add News**.
The News page appears.
 - b. Enter text in the News Entry field.
 - c. Click **Add News**.
5. To edit a news item:
 - a. Select the news item.
 - b. Edit the News Entry.
 - c. Click **Apply Changes**.
6. To delete a news item:
 - a. Select the news item.

- b. Click **Delete**.

4.9.9 Managing Links

Use the Links page to share links within Team Development.

- [Adding Links to the Links Page](#) (page 4-29)
- [Editing or Deleting a Link](#) (page 4-29)

4.9.9.1 Adding Links to the Links Page

To access the links page:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Manage Links**.
The Links page appears.
3. Use the Search bar to filter the display.
4. To add a link:
 - a. Click **Create Link**.
The Links page appears.
 - b. 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 **Create Link**.

4.9.9.2 Editing or Deleting a Link

To edit or delete a link:

1. On the Workspace home page, click the **Team Development** icon.
2. Under Utilities, click **Manage Links**.
The Links page appears.
3. To edit a link:
 - a. Click the **Edit** icon.
The Links page appears.
 - b. Edit the appropriate fields.
 - c. Click **Apply Changes**.
4. To delete a new link:
 - a. Click the **Edit** icon.
The Links page appears.
 - b. Click **Delete**.

5

Creating Websheet Applications

Websheet applications are interactive web pages that combine text with data. This section describes how to create Websheets.

Topics:

- [About Websheet Applications](#) (page 5-1)
- [Creating a Websheet Application](#) (page 5-2)
- [Copying an Existing Websheet Page](#) (page 5-4)
- [Running a Websheet](#) (page 5-4)
- [Editing Page Details](#) (page 5-6)
- [Deleting a Websheet Application](#) (page 5-7)
- [Viewing the Websheet Dashboard](#) (page 5-7)
- [Monitoring Websheet Activity](#) (page 5-7)
- [Understanding Websheet Properties](#) (page 5-8)
- [Controlling Websheet Access](#) (page 5-12)



See Also:

- "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*

5.1 About Websheet Applications

Websheet applications are interactive web pages that combine text with data. These applications are highly dynamic and defined by their users. Websheet applications include navigation controls, search capabilities, and the ability to add annotations such as files, notes, and tags. Websheet applications can be secured using access control lists and several built-in authentication models.

 **See Also:**

- ["What is the Difference Between a Database and Websheet Application? \(page 2-2\)"](#)
- ["Understanding AnyCo IT Department Websheet \(page 3-19\)"](#)
- ["Using Websheets" in *Oracle Application Express End User Guide*](#)

5.2 Creating a Websheet Application

This section describes how to create a Websheet application. You create a Websheet application by running the Create Application wizard. After you create a Websheet page, you add content either as text or data.

- [Creating a Websheet from App Builder \(page 5-2\)](#)
- [Creating a Websheet Application from a Running Websheet \(page 5-3\)](#)

 **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*](#)

5.2.1 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.
3. For the application type, 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 name to identify the application.
 - c. Allow SQL - Determines whether you can define SQL tags and SQL reports. Select **Yes** or **No**. By default, this attribute is disabled and set to **No**.
 - d. 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 packaged 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.

Your Websheet application displays in a new window.

7. Add Websheet content.

 **See Also:**

["About Publishing the Websheet Application URL \(page 23-39\)"](#) and ["Using Websheets"](#) in *Oracle Application Express End User Guide*

5.2.2 Creating a Websheet Application from a Running Websheet

To create a Websheet application from a running Websheet:

1. Run a Websheet application. See ["Running a Websheet \(page 5-4\)"](#).
2. From the Create menu, click **New Websheet**.
3. For Create Websheet:
 - a. Websheet - Identifies 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. Description - Enter a description of your application.
 - d. Click **Next**.
4. Confirm your selections and click **Create**.
5. To view a rendered version of your Websheet application, click **Run Websheet**.
Your Websheet application appears in a new window.
6. Add Websheet content.

 **See Also:**

- ["About Packaged Applications \(page 3-1\)"](#)
- ["Using Worksheets" in *Oracle Application Express End User Guide*](#)
- ["About Publishing the Worksheet Application URL \(page 23-39\)"](#)

5.3 Copying an Existing Worksheet Page

To copy an existing Worksheet page:

1. Run the Worksheet 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 \(page 5-6\)](#)."

 **See Also:**

["Running a Worksheet \(page 5-4\)"](#)

5.4 Running a Worksheet

To view a rendered version of your application, you run or submit it to the Application Express engine.

- [Running a Worksheet from the App Builder Home Page \(page 5-5\)](#)
- [Running a Worksheet from the Application Home Page \(page 5-5\)](#)
- [Determining a Worksheet URL \(page 5-6\)](#)

 **See Also:**

"Using Worksheets" in *Oracle Application Express End User Guide* and "[About Publishing the Worksheet Application URL](#) (page 23-39)"

5.4.1 Running a Worksheet from the **App Builder** Home Page

To run a Worksheet 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 Worksheet application in the Applications list.
4. Click the **Run** icon in the far right column.
5. Enter your Worksheet 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 packaged 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 Worksheet application appears in a new window.

5.4.2 Running a Worksheet 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 Worksheet application.
3. Click the **Run** button.
4. Enter your Worksheet 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 packaged 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.

5.4.3 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](#) (page 23-39)."

5.5 Editing Page Details

On Page Details, you can 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.
 - Parent Page - Identifies the parent page. The defined parent page is used to construct breadcrumbs and navigation between pages.
 - 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](#) (page 5-4)"

5.6 Deleting a Websheet Application

You 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 \(page 5-8\)"](#)

5.7 Viewing the Websheet Dashboard

The Websheet dashboard displays the details about the current Websheet application including recent changes, top users, and top pages.

To view the Websheet Dashboard:

1. Run the Websheet. See ["Running a Websheet \(page 5-4\)"](#).
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.

5.8 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](#) (page 8-92)."



See Also:

"[Running a Websheet](#) (page 5-4)"

5.9 Understanding Websheet Properties

You can access the Websheet Properties page from inside or outside a running Websheet. Use the Websheet Properties page to control general application defaults and behavior. The home page identifies the home page users land on when running the Websheet. The name of the application is displayed within the breadcrumb.

- [Editing Websheet Properties](#) (page 5-8)
- [Websheet Properties](#) (page 5-9)



See Also:

"[Deleting a Websheet Application](#) (page 5-7)"

5.9.1 Editing Websheet Properties

To access the Websheet Properties page:

1. Log in to and run the Websheet.
2. From the Administration menu, select **Websheet Properties**.
The Websheet Properties appears.
3. To view a section, click the greater than sign (>) to the left of the section title.
4. Edit the appropriate properties and click **Apply Changes**.

**See Also:**["Running a Websheet \(page 5-4\)"](#)

5.9.2 Websheet Properties

This section describes the attributes on the Websheet Properties page.

- [Websheet \(page 5-9\)](#)
- [Details \(page 5-10\)](#)
- [Logo \(page 5-10\)](#)
- [Style \(page 5-10\)](#)
- [Authentication \(page 5-11\)](#)
- [Authorization \(page 5-11\)](#)
- [SQL \(page 5-12\)](#)
- [Websheet \(page 5-9\)](#)
- [Details \(page 5-10\)](#)
- [Logo \(page 5-10\)](#)
- [Style \(page 5-10\)](#)
- [Authentication \(page 5-11\)](#)
- [Authorization \(page 5-11\)](#)
- [SQL \(page 5-12\)](#)

5.9.2.1 Websheet

Use the Websheet section to control general Websheet application properties. [Table 5-1 \(page 5-9\)](#) describes Application attributes.

Table 5-1 Websheet Attributes

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.

Table 5-1 (Cont.) Websheet Attributes

Attribute	Descriptions
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.

5.9.2.2 Details

Use the Details section to edit the application description and login message. [Table 5-2](#) (page 5-10) describes Details attributes.

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.

5.9.2.3 Logo

Use the Logo section to create a logo for this Websheet application. This logo displays at the top of the Websheet. [Table 5-3](#) (page 5-10) describes Logo attributes.

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.

5.9.2.4 Style

Use the Style section to override default application style and use a custom CSS.

5.9.2.5 Authentication

Use the Authentication section to select an authentication scheme for this Websheet. [Table 5-4](#) (page 5-11) describes Authentication attributes.

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 (page 5-12)."
Logout URL	<p>Specify a URL to become this application's Logout URL attribute. This attribute is referenced in logout links as follows:</p> <p>&LOGOUT_URL.</p> <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>
Cookie Name	<p>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.</p>
Cookie Path	<p>Path attribute for the session cookie.</p>
Cookie Domain	<p>Domain attribute for the session cookie.</p>
Secure	<p>Options include:</p> <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.

5.9.2.6 Authorization

Use the Authorization section to control who can log in to a Websheet application. [Table 5-5](#) (page 5-12) describes Application attributes.

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](#) (page 5-12)."

5.9.2.7 SQL

Use the SQL section to 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*.

5.10 Controlling Websheet Access

This section describes how to control access to a Websheet application.

- [How Websheet Access Works](#) (page 5-13)
- [Specifying the Access Control List Type](#) (page 5-13)
- [Creating Access Control List Entries](#) (page 5-14)

5.10.1 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.

To learn more about user roles and privileges, click **Help** and then select **Access Control**.

5.10.2 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. Click the **Websheet Applications** tab.
3. Click the **View Icons** button.
4. Locate the Websheet and click **Edit**.
The Websheet Properties appears.
5. Scroll down to Authorization.
6. From Access Control List Type, select **Custom**.
7. Click **Apply Changes**.



See Also:

["Authentication \(page 5-11\)"](#)

5.10.3 Creating Access Control List Entries

To create access control list entries:

1. Log in to and 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 - Enter the username of the user for whom you want to control access.
 - b. Privilege - Identify the user's privileges:
 - 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 \(page 5-4\)"](#)

6

Creating Database Applications

Developers create applications in Oracle Application Express using the Create Application Wizard. 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.

Topics:

- [Using the Create Application Wizard](#) (page 6-1)
- [Creating a Database Application](#) (page 6-4)
- [Creating Mobile Applications](#) (page 6-11)
- [Deleting an Application](#) (page 6-15)
- [Managing Application Attributes](#) (page 6-16)
- [Managing Pages in a Database Application](#) (page 6-44)
- [Using the Find Icon](#) (page 6-55)
- [Creating Page Groups](#) (page 6-61)
- [Locking and Unlocking a Page](#) (page 6-64)
- [Adding Developer Comments](#) (page 6-68)
- [Adding Database Application Comments](#) (page 6-71)
- [Managing Application Groups](#) (page 6-71)
- [Using Application Utilities](#) (page 6-74)

See Also:

["Managing User Interface Design \(page 2-5\)"](#) and ["What is the Difference Between a Database and Websheet Application? \(page 2-2\)"](#)

6.1 Using the Create Application Wizard

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.

- [About the Create Application Wizard](#) (page 6-2)
- [Available Page Types When Running the Create Application Wizard](#) (page 6-2)



See Also:

"[Managing User Interface Design](#) (page 2-5)," "[Creating a Database Application](#) (page 6-4)," and "[Creating Mobile Applications](#) (page 6-11)"

6.1.1 About the Create Application Wizard

The Create Application Wizard enables developers to design and quickly create basic Oracle Application Express applications. By using the Create Application Wizard, you can create a complete application containing multiple pages including a variety of different reports and forms. After you create an application using this wizard, you can modify pages and add additional pages using the Create Page Wizard.

You can use the Create Application Wizard to create an application with just a blank page or pages based on SQL queries on existing database tables. Applications based on tables can consist of a simple report, a form and report, or an editable interactive grid. You can create SQL queries by manually typing SQL, or by using the graphical user interface of Query Builder.

At the workspace level, you have the option of setting defaults that are used for applications built in that workspace. Setting these defaults enables you to exit the wizard without paging through each step. When you exit the wizard, the application you create uses the values you specified as defaults.



See Also:

- "[About Adding Pages to a Database Application](#) (page 6-45)"
- "Building Queries with Query Builder" in *Oracle Application Express SQL Workshop Guide*
- "Leveraging App Builder Defaults" in *Oracle Application Express Administration Guide*

6.1.2 Available Page Types When Running the Create Application Wizard

When you run the Create Application Wizard, you choose a target user interface. The user interface you select determines the available page types. Not all page types are supported by all user interfaces.

Supported Page Types for Desktop Applications

If you are creating a desktop application, the Create Application Wizard supports the following page types.

Table 6-1 Create Application Wizard - Supported Page Types for Desktop Applications

Page Type	Description	To Learn More
Blank	Creates a page with no built-in functionality.	Not applicable.
Report	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, interactive report , interactive grid , or classic report).	See " Developing Reports (page 8-1)"
Report and Form	You choose a table on which to build a two page report and form combination. You then choose the report type (that is, interactive report , interactive grid , or classic report). On the first page, users select a row to update. On the second page, users can add a record, or update or delete an existing record.	See " Developing Reports (page 8-1)"
Form	Creates a form to update a single row in a database table. You choose a table on which to build the form.	See " Developing Forms (page 9-1)"
Editable Interactive Grid	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. You choose a table on which to build the interactive grid.	See " Managing Interactive Grids (page 8-13)"
Master Detail	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.	See " Managing Master Detail Forms (page 9-14)"
Chart	Creates a page with a chart to represent the result of a SQL query.	See " Creating Charts (page 10-23)"

Supported Page Types for Mobile Applications

If you are creating a mobile application, the Create Application Wizard supports the following page types.

Table 6-2 Create Application Wizard - Supported Page Types for Mobile Applications

Page Type	Description	To Learn More
Blank	Creates a page with no built-in functionality.	Not applicable.

Table 6-2 (Cont.) Create Application Wizard - Supported Page Types for Mobile Applications

Page Type	Description	To Learn More
List View	Creates a responsive report for mobile applications. The report works by collapsing the table columns into a stacked presentation that looks like blocks of label and data pairs for each row. This switch occurs when there is not enough space available to display the report horizontally.	See " Developing Reports (page 8-1)"
Column Toggle Report	Creates a responsive report for mobile applications. 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 display at narrower widths (for example, columns with priority 6 will be hidden first and columns with priority 1 will be hidden last.) The report includes a Columns button which enables end users to select which columns they want to view.	See " Developing Reports (page 8-1)"
Reflow Report	Creates a responsive report for mobile applications. When there is not enough space available to display the report horizontally, the report works by collapsing the table columns into a stacked presentation that looks like blocks of label and data pairs for each row.	See " Developing Reports (page 8-1)"
List View and Form	Optimized to display data and provide easy navigation on Smartphones. Builds a two page List view and form combination. On the first page, users select a row to update from the List View. On the second page, users can update the selected table or view.	See " Developing Reports (page 8-1)"
Form	Creates a form to update a single row in a database table. You choose a table on which to build the form.	See " Developing Forms (page 9-1)"
Chart	Creates a page with a chart to represent the result of a SQL query.	See " Creating Charts (page 10-23)"

6.2 Creating a Database Application

You can create a database application based on a table, query, or drill-down query by selecting **Database** in the Create Application Wizard. A key decision point when running the Create Application Wizard is choosing a target user interface. Based on the user interface you select, the wizard displays the appropriate page types, attributes, and themes.

- [Creating a Database Application Based on a Table, Query, or Drill-Down Query](#) (page 6-5)
- [Creating a Desktop Database Application from a Spreadsheet](#) (page 6-8)

- [Copying a Database Application](#) (page 6-9)

 **See Also:**

- ["What is the Difference Between a Database and Websheet Application?"](#) (page 2-2)"
- ["Managing User Interface Design](#) (page 2-5)"
- ["Creating Mobile Applications](#) (page 6-11)"
- ["Deleting an Application](#) (page 6-15)"

6.2.1 Creating a Database Application Based on a Table, Query, or Drill-Down Query

To create a database application based on a table, query, or drill-down query:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Select the application type, select **Desktop** and click **Next**.
4. For Name:
 - a. Schema - Each application obtains its privileges by parsing all SQL as a specific database schema. Identify the database schema owner.
 - b. Name - Enter the name used to identify the application to developers.
 - c. Application - Specify an integer value that identifies the application. The application ID must be unique within your Application Express instance. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - d. Theme - Select a theme. Themes are collections of templates used to define the layout and style of an entire application.

To view available themes, click **Select Theme**. Standard themes display. To view additional themes, make a selection from the Show list. When you select a theme the application automatically subscribes to it.
 - e. Theme Style - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51.
 - f. Click **Next**.

 **Tip:**

To create an instant application that includes just a login page and blank page, click the **Create Application** button.

Next, add pages to your application.

5. For Pages, click **Add Page**.

- a. Select Page Type - Select the type of page you want to add. See "[Available Page Types When Running the Create Application Wizard](#) (page 6-2)."

Based on the Page Type, the wizard prompts you for different information, such as selecting a table name or report implementation. To learn more, see field-level Help. **Action** displays the currently selected page type. For each selection, the wizard prompts you for different types of information, such as selecting a table name or report implementation. To learn more, see field-level Help.

- b. Click **Add Page**.

The page (or pages) appear at the top of the page. To delete a page, click the **Delete** icon (the X in the far right column).

- c. Repeat the previous steps until all pages have been added.



Tip:

To exit the wizard early and use App Builder Defaults, click the **Create Application** button.

- d. Click **Next**

6. For Shared Components, indicate whether to copy shared components from another application. Select **Yes** or **No**. If you select **Yes**, select where to copy them from and which components to import.

Shared components are common application elements that can be displayed or applied across multiple pages in an application. To save time or maintain consistency between applications, copy the shared components from an existing application. The types of shared components you can copy include: user interface themes, lists of values, navigation bar entries, authorization schemes and authentication schemes. If copying the theme from another application, you are to subscribe the theme.

7. For Attributes:

- a. Authentication Scheme - Determine how to establish user identity before accessing an application. Options include:
 - **Application Express Accounts** - Utilizes the user account credentials created and maintained with the Application Express Administration Services application. These are the accounts you use to log in to the Application Express development environment. You can also create accounts in this user account repository for end users of your applications.
 - **No Authentication** - Also known as database authentication. This option enables users to access your application using the account credentials stored in the `mod_plsql` DAD definition. This authentication scheme requires that a database user (schema) exists in the local database. In most cases this results in users not having to login when accessing your application and is quickest way to create a "public" application.
 - **Database Account** - Requires users log in to an application by entering a database schema name (or username) and password. This account information is managed entirely within the Oracle database.
- b. Language - Select the primary language for this application.

This attribute identifies the language in which an application is developed. This language is the base language from which all translations are made.

User Language Preference 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, see field-level Help.

 **Tip:**

You can alter the Language and User Language Preference Derived From attributes later on the Edit Globalization attributes page. See "[Configuring Globalization Attributes](#) (page 6-32)."

- c. Date Format - Specifies 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 within the application. This value can be a literal string containing a valid Oracle date format mask or an item reference through substitution syntax. If no value is specified, the default date format is derived from the database session at runtime. If supplied, this is also used as the date format for any items resulting from columns of type `DATE`.

- d. Date Time Format - Specifies 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.

- e. Timestamp Format - Specifies the timestamp format to be used in your application.

Use this timestamp format to alter the `NLS_TIMESTAMP_FORMAT` database session setting before showing or submitting any page within the application. This value can be a literal string containing a valid Oracle timestamp format mask or an item reference through substitution syntax. If no value is specified, the default timestamp format is derived from the database session at runtime.

- f. Timestamp Time Zone Format - Specifies the timestamp with time zone format to be used in your application.

Use this timestamp with time zone format to alter the `NLS_TIMESTAMP_TZ_FORMAT` database session setting before showing or submitting any page within the application. This value can be a literal string containing a valid Oracle timestamp with time zone format mask or an item reference through substitution syntax. If no value is specified, the default timestamp with time zone format is derived from the database session at runtime.

- g. Click **Next**.

8. Confirm your selections and click **Create Application**.

 **See Also:**

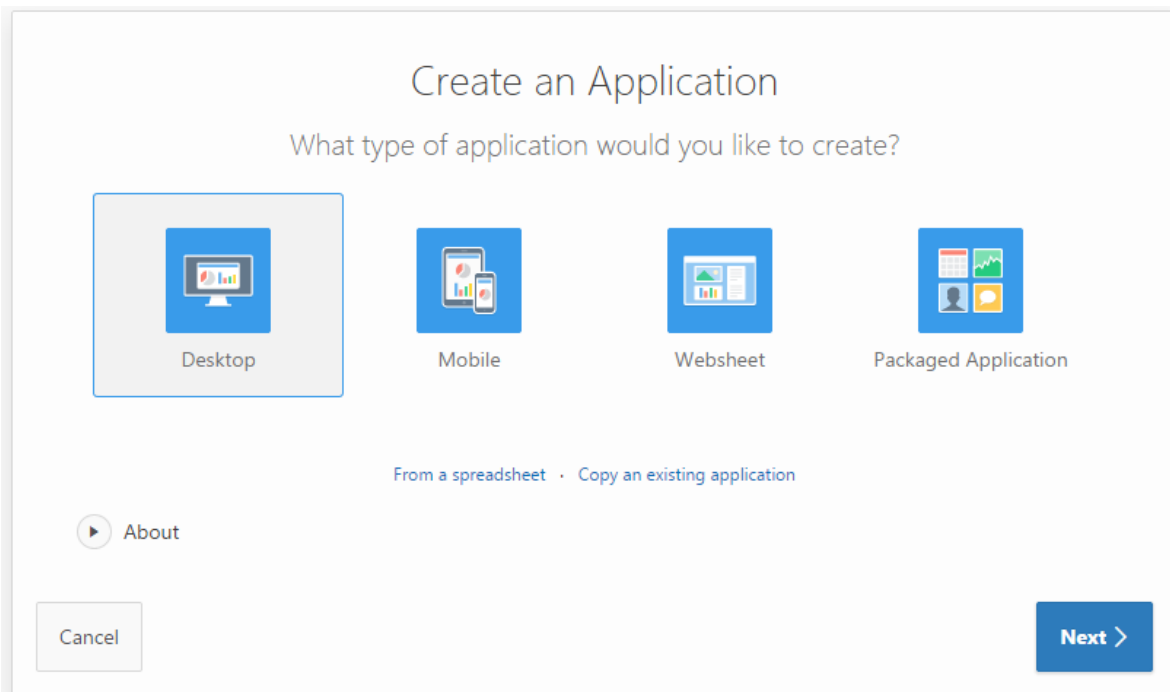
- ["Using Themes \(page 11-1\)"](#)
- ["Using Theme Styles \(page 11-3\)"](#)
- ["Understanding Theme Subscriptions \(page 11-17\)"](#)
- ["Establishing User Identity Through Authentication \(page 19-45\)"](#)

6.2.2 Creating a Desktop Database Application from a Spreadsheet

You can create a desktop application based on spreadsheet data by selecting **From Spreadsheet** in the Create Application Wizard.

To create an application from spreadsheet data:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. Select **From a spreadsheet**.



4. For Load Method, specify how spreadsheet data is imported:
 - **Upload file, comma separated (*.csv) or tab delimited.** Specify the following and click **Next**:
 - Text File - Click **Choose File** to locate the file to be uploaded.
 - Separator - Specify the column separator character. Use \t for tab separators.

- 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.
 - First row contains column names - Select this option if your data contains column names in the first row.
 - File Character Set - Choose the character set in which the text file is encoded.
- **Copy and paste.** Copy and paste the tab delimited data you want to import.
5. For Table Properties:
 - a. Schema - Identify the database schema owner.
 - b. Table - Enter a name for the table being created.
 - c. Set Table Properties - Review the table properties. Modify the data types or enter format masks. Specify whether to include a column by selecting **Yes** or **No** from the Upload list.
 - d. Click **Next**.
 6. For Application Options:
 - a. Application Name - Enter an alphanumeric name for this application.
 - b. Report Type - Select one of the following
 - Interactive Grid - Creates an interactive grid.
 - Interactive Report - Creates an interactive report.
 - Classic - Creates a SQL report.
 - c. For Page Type - Select the type of page to generate. The options that display depend upon the selected Report Type.

 **Tip:**

If the Report Type is **Interactive Grid**, you can create either a non-editable interactive grid (report) page and a standard form page, or an interactive grid on a single page. All other report types create a report page and a form page.

7. Click **Create Application**.

 **See Also:**

"[Using Themes](#) (page 11-1)" and "[Using Theme Styles](#) (page 11-3)."

6.2.3 Copying a Database Application

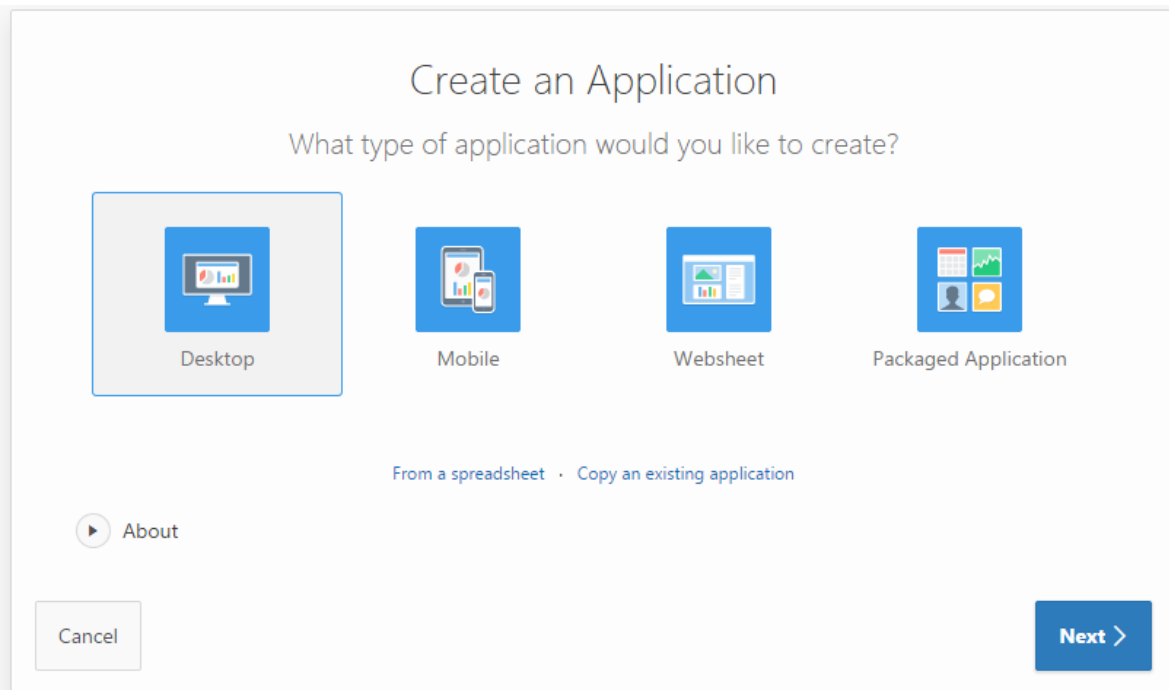
You can create a copy of an existing application by running the Create Application Wizard or by selecting the application and then **Copy this Application** on the Application home page.

- [Copying a Database Application Using the Create Application Wizard](#) (page 6-10)
- [Copying an Application from the Application Home Page](#) (page 6-11)

6.2.3.1 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 an existing 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**.

6.2.3.2 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**.

6.3 Creating Mobile Applications

Because mobile applications are designed to run specifically on mobile devices, Oracle recommends building them with minimal, semantic HTML that is optimized for mobile connections.

- [About jQuery Mobile Support](#) (page 6-11)
- [Creating a Mobile Application](#) (page 6-12)

See Also:

- ["Managing User Interface Design](#) (page 2-5)"
- ["About Responsive Design](#) (page 11-2)"
- ["Deleting an Application](#) (page 6-15)"

6.3.1 About jQuery Mobile Support

App Builder

To support mobile application development, Oracle Application Express incorporates jQuery Mobile. jQuery Mobile-based web pages use a lightweight framework that enables you to create compact and minimalistic web sites that include only a few

images and CSS files. Using jQuery Mobile, these applications can run on the following mobile operating systems: iOS, Android, Blackberry, and Windows Mobile.

jQuery Mobile supports mobile device-specific events, such as orientation change and touch events. You easily change the look and feel of jQuery Mobile-based applications by modifying the CSS using tools such as ThemeRoller.

 **Tip:**

To learn more about ThemeRoller, go to <http://themeroller.jquerymobile.com>.

To learn more about jQuery Mobile, go to <http://jquerymobile.com>.

 **See Also:**

"[Managing User Interface Design](#) (page 2-5)"

6.3.2 Creating a Mobile Application

To create a mobile application:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.
3. For application type, select **Mobile** and click **Next**.
4. For Name:
 - a. Schema - Each application obtains its privileges by parsing all SQL as a specific database schema. Identify the database schema owner.
 - b. Name - Enter the name used to identify the application to developers.
 - c. Application - Specify an integer value that identifies the application. The application ID must be unique within your Application Express instance. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - d. Theme - Select a theme. Themes are collections of templates used to define the layout and style of an entire application.
 - e. Theme Style - Select a Theme Style for your application. Theme styles define the CSS for your application's theme and can change the UI of your application.
 - f. Click **Next**

 **Tip:**

To create an instant application consisting that includes just a login page and blank page, click the **Create Application** button.

Next, add pages to your application.

5. For Pages, click **Add Page**.

In the Create Application - Add Page dialog:

- a. For Select Page Type, select the type of page you want to add. See "[Available Page Types When Running the Create Application Wizard](#) (page 6-2)."

The available page types depends upon the User Interface you select in the previous step.

Action displays the currently selected page type.

For each selection, the wizard prompts you for different types of information, such as selecting a table name or report implementation. To learn more, see field-level Help.

- b. Click **Add Page**.

The page (or pages) appear at the top of the page. To delete a page, click the **Delete** icon.

- c. Repeat the previous steps until all pages have been added.

- d. Click **Next**.

 **Tip:**

To exit this wizard early and use App Builder Defaults, click the **Create Application** button. To configure App Builder Defaults, see "Leveraging App Builder Defaults" in *Oracle Application Express Administration Guide*.

6. For Shared Components, indicate whether to copy shared components from another application. Select **Yes** or **No**. If you select Yes, select where to copy them from and which components to import.

Shared components are common application elements that can be displayed or applied across multiple pages in an application. To save time or maintain consistency between applications, copy the shared components from an existing application. The types of shared components you can copy include: user interface themes, lists of values, navigation bar entries, authorization schemes and authentication schemes.

7. For Attributes:

- a. Authentication Scheme - Determine how to establish user identity before accessing an application.

To learn more, see field-level Help.

- b. Language - Select the primary language for this application.

This attribute identifies the language in which an application is developed. This language is the base language from which all translations are made.

- c. User Language Preference 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 items. To learn more, see field-level Help.

 **Tip:**

You can alter the Language and User Language Preference Derived From attributes later on the Edit Globalization attributes page.

- d. Date Format - Specifies the date format to be used in the application.

Use Date Format to alter the `NLS_DATE_FORMAT` database session setting before showing or submitting any page within the application. This value can be a literal string containing a valid Oracle date format mask or an item reference through substitution syntax. If no value is specified, the default date format is derived from the database session at runtime. If supplied, this is also used as the date format for any items resulting from columns of type `DATE`.

- e. Date Time Format - Specifies the date time format to be used in the application.

You can reference the Date Time Format 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.

- f. Timestamp Format - Specifies the timestamp format to be used in your application.

Use Timestamp Format to alter the `NLS_TIMESTAMP_FORMAT` database session setting before showing or submitting any page within the application. This value can be a literal string containing a valid Oracle timestamp format mask or an item reference through substitution syntax. If no value is specified, the default timestamp format is derived from the database session at runtime.

- g. Timestamp Time Zone Format - Specifies the timestamp with time zone format to be used in your application.

Use Timestamp Time Zone Format to alter the `NLS_TIMESTAMP_TZ_FORMAT` database session setting before showing or submitting any page within the application. This value can be a literal string containing a valid Oracle timestamp with time zone format mask or an item reference through substitution syntax. If no value is specified, the default timestamp with time zone format is derived from the database session at runtime.

- h. Click **Next**.

- 8. Confirm your selections and click **Create Application**.

 **See Also:**

- ["Using Themes \(page 11-1\)"](#)
- ["Using Theme Styles \(page 11-3\)"](#)
- ["Understanding Theme Subscriptions \(page 11-17\)"](#)
- ["Establishing User Identity Through Authentication \(page 19-45\)"](#)

6.4 Deleting an Application

Developers can delete an application from within App Builder, or while editing application attributes. When you delete an application, you also delete all defined components (reports, charts, or forms), page controls (buttons, items, list of values), and shared components (breadcrumbs, lists, and tabs, but not user interface defaults).

- [Deleting an Application from the Application Home Page \(page 6-15\)](#)
- [Deleting an Application from the Edit Application Definition Page \(page 6-15\)](#)

 **See Also:**

["Deinstalling Supporting Objects \(page 23-15\)"](#) and ["Removing an Installed Packaged Application \(page 3-6\)"](#)

6.4.1 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 appears, 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**.

6.4.2 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 \(page 6-16\)"](#)

6.5 Managing Application Attributes

Application attributes control the behavior of to an entire application and are divided into the categories: Definition, Security, Globalization, and User Interface.

- [Editing Application Attributes \(page 6-16\)](#)
- [Configuring Security Attributes \(page 6-23\)](#)
- [Configuring Globalization Attributes \(page 6-32\)](#)
- [Managing the Application User Interface \(page 6-36\)](#)



See Also:

["How to Create a Custom Packaged Application \(page 23-8\)"](#) for information on using the Supporting Objects utility to create a packaged application

6.5.1 Editing Application Attributes

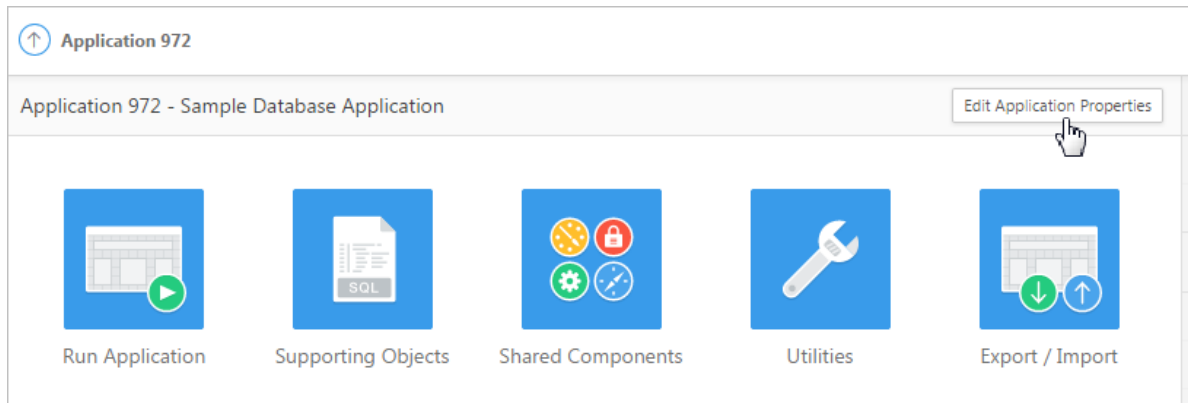
Edit application attributes on the Edit Application Definition page to alter the application name, modify various application properties and behavior, control application availability, configure error handling, and define substitution strings.

- [Accessing the Edit Application Definition Page \(page 6-16\)](#)
- [Edit Application Definition Page \(page 6-17\)](#)

6.5.1.1 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.
3. Click the **Edit Application Properties** button to the right of the application name.



The Edit Application Definition page appears.

4. Edit the appropriate attributes and click **Apply Changes**.

6.5.1.2 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](#) (page 6-17)
- [Properties](#) (page 6-18)
- [Availability](#) (page 6-20)
- [Error Handling](#) (page 6-21)
- [Global Notification](#) (page 6-22)
- [Substitutions](#) (page 6-23)
- [Build Options](#) (page 6-23)

6.5.1.2.1 Name

Use Name attributes to define basic characteristics of your application, including the application name, an optional alphanumeric alias, and a version number. [Table 6-3](#) (page 6-17) describes all Name attributes.

Table 6-3 Application Definition, Name

Attribute	Description
Name	A short descriptive name for the application to distinguish it from other applications in your development environment.

Table 6-3 (Cont.) Application Definition, Name

Attribute	Description
Application Alias	<p>Assigns an alternate alphanumeric application identifier. You can use this identifier for the application ID.</p> <p>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:</p> <ul style="list-style-type: none"> <code>f?p=105:1</code> <code>f?p=myapp:1</code> <p>See Also: "About Using f?p Syntax to Link Pages (page 2-17)"</p>
Version	<p>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:</p> <ul style="list-style-type: none"> <code>YYYY.MM.DD</code> <code>MM.DD.YYYY</code> <code>DD.MM.YYYY</code> <p>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.</p>
Application Group	<p>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.</p> <p>See Also: "Managing Application Groups (page 6-71)"</p>

6.5.1.2.2 Properties

Use the Properties section to configure logging, debug behavior, feedback, compatibility, email from address, and the application proxy server. [Table 6-4](#) (page 6-18) describes all Name attributes.

Table 6-4 Application Definition, Properties

Attribute	Description
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> <p>See Also: "Enabling Application Activity" in <i>Logging in Oracle Application Express Administration Guide</i>.</p>

Table 6-4 (Cont.) Application Definition, Properties

Attribute	Description
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. • No - Prevents end users from running an application in debug mode. <p>Running an application in debug mode is useful when an application is under development. For production applications, Oracle recommends disabling debugging and thus prevent 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>
Allow Feedback	<p>Enables support for end user feedback for this application. Select Yes or No.</p> <p>If 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.</p> <p>See Also: "Managing Feedback (page 4-20)"</p>
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>
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 <code>APP_EMAIL</code>. 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>You can also specify the from address on the Interactive Report Attributes page. See "Editing Classic Report Attributes in Component View (page 8-112)."</p>
Proxy Server	<p>Specify a proxy server.</p> <p>For example, you 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>

6.5.1.2.3 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. To learn more, see [Table 6-5](#) (page 6-20).

Table 6-5 Application Definition, Availability

Attribute	Description
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 Also:</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 (page 19-39)"
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 Also: "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>

Table 6-5 (Cont.) Application Definition, Availability

Attribute	Description
Restrict to comma separated user list (status must equal Restricted Access)	<p>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.</p> <p>To use this attribute:</p> <ol style="list-style-type: none"> 1. From the Status list, select a restricted status. 2. Enter a comma-delimited list of users who can run the application in the field provided. 3. Click Apply Changes.

6.5.1.2.4 Error Handling

Use the Error Handling attributes described in [Table 6-6](#) (page 6-21) to control or modify how an application logs errors.

 **Tip:**

Error handling functions specified here are overridden by similar page-level attributes.

Table 6-6 Application Definition, Error Handling

Attribute	Description
Default Error Display Location	<p>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:</p> <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 #NOTIFICATION_MESSAGE# template substitution string when an error occurs on the page.

Table 6-6 (Cont.) Application Definition, Error Handling

Attribute	Description
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>See Also: <code>apex_error</code> in <i>Oracle Application Express API Reference</i></p> <p>Note: Error handling specified at the page-level overwrites any error handling function specified here.</p>

6.5.1.2.5 Global Notification

Use the Global Notification attribute to communicate system status to application users. For example, you can use this attribute to notify users of scheduled downtime, or communicate other messages regarding application availability. If the page templates used in your application contain the #GLOBAL_NOTIFICATION# substitution string, the text entered here displays in that string's place.

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:

"[Understanding Substitution Strings](#) (page 2-26)" and "[Page Templates](#) (page 11-59)"

6.5.1.2.6 Substitutions

Use Substitutions section 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:**

["Understanding Substitution Strings \(page 2-26\)"](#)

6.5.1.2.7 Build Options

The Build Options section displays existing build options. 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 considers it at runtime. However, if you specify an attribute to be excluded, then the Application Express engine treats it as if it did not exist.

Do not specify a build option unless you plan to exclude that object from specific installations.

**See Also:**

["Using Build Options to Control Configuration \(page 23-40\)"](#)

6.5.2 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 \(page 6-24\)](#)
- [Security Attributes Page \(page 6-24\)](#)

**See Also:**

["Managing Application Security \(page 19-1\)"](#)

6.5.2.1 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.
3. Click **Shared Components**.
The Shared Components page appears.
4. Under Security, click **Security Attributes**.
The Edit Security Attributes page appears.
5. Edit the appropriate attributes and click **Apply Changes**.

Tip:

You can also access to the Edit Security Attributes by navigating to the Edit Application Definition and then clicking the **Security** tab. See "[Accessing the Edit Application Definition Page](#) (page 6-16)."

6.5.2.2 Security Attributes Page

The Edit Security Attributes page is divided into the following sections:

Note:

Required values are marked with a red asterisk (*).

- [Authentication](#) (page 6-24)
- [Authorization](#) (page 6-25)
- [Session Management](#) (page 6-26)
- [Session State Protection](#) (page 6-29)
- [Browser Security](#) (page 6-30)
- [Database Session](#) (page 6-31)

6.5.2.2.1 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 6-7](#) (page 6-25) describes the attributes available under Authentication.

Table 6-7 Authentication Attributes

Attribute	Descriptions
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 Also: "HOME_LINK (page 2-40)" and "Understanding Conditional Rendering and Processing (page 2-10)"</p>
Authentication Scheme	<p>Select from the authentication schemes defined for the application. To create an authentication scheme, click Define Authentication Schemes.</p> <p>See Also: "How Authentication Works (page 19-46)" and "Creating an Authentication Scheme (page 19-47)"</p>

6.5.2.2.2 Authorization

Application authorization schemes control access to all pages within an application. Unauthorized access to the application, regardless of which page is requested, will cause an error page to display. [Table 6-8](#) (page 6-25) describes the attributes available under Authorization.

Table 6-8 Authorization Attributes

Attribute	Descriptions
Authorization Scheme	<p>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.</p>
Define Authorization Schemes	<p>Click Define Authorization Schemes to create an authorization scheme.</p>

Table 6-8 (Cont.) Authorization Attributes

Attribute	Descriptions
Run on Public Pages	<p>Controls whether the application-level authorization scheme is checked on public pages (that is, pages that do not require authorization). Options include:</p> <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. <p>Application-level authorization is never evaluated on the login page (independent of Run on Public Pages), because authorizations typically are dependent on the username.</p>

**See Also:**

["Attaching an Authorization Scheme to an Application \(page 19-64\)"](#)

6.5.2.2.3 Session Management

Use Session Timeout attributes to reduce exposure to abandoned computers with an open web browser by application. [Table 6-9](#) (page 6-27) describes the attributes available under Session Timeout.

Table 6-9 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 (page 19-7)."</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: <ul style="list-style-type: none"> 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> <p>See Also: "Security (page 24-19)" to learn more about overriding this behavior at the page-level.</p>

Table 6-9 (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. <p>The session duration may be superseded by the operation of the job that runs every eight hours which deletes sessions older than 12 hours.</p>
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. • Set the value to 0 to prevent session idle time checks from being performed. • 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](#) (page 19-8)" and "Configuring Session Timeout" in *Oracle Application Express Administration Guide*

6.5.2.2.4 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. [Table 6-10](#) (page 6-29) describes the attributes available under Session State Protection.

Table 6-10 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:**

["Understanding Session State Protection \(page 19-26\)"](#)

6.5.2.2.5 Browser Security

[Table 6-11](#) (page 6-30) describes the attributes available under Browser Security.

Table 6-11 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 <code>AL32UTF8</code>

Table 6-11 (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.

6.5.2.2.6 Database Session

[Table 6-12](#) (page 6-31) describes the attributes available under Database Session.

Table 6-12 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.
Runtime API Usage	<p>Control how this application can access Oracle Application Express APIs that modify applications and workspace data, while it is running. Options include:</p> <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](#) (page 19-59)" and *Oracle Label Security Administrator's Guide*

6.5.3 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](#) (page 6-32)
- [Edit Globalization Attributes Page](#) (page 6-33)

**See Also:**

"[Managing Application Globalization](#) (page 20-1)"

6.5.3.1 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](#) (page 6-16)."

6.5.3.2 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](#) (page 6-33)
- [Application Language Derived From](#) (page 6-33)
- [Document Direction](#) (page 6-34)
- [Application Date Format](#) (page 6-34)
- [Application Date Time Format](#) (page 6-34)
- [Application Timestamp Format](#) (page 6-34)
- [Application Timestamp Time Zone Format](#) (page 6-34)
- [Character Value Comparison](#) (page 6-35)
- [Character Value Comparison Behavior](#) (page 6-35)
- [Automatic Time Zone](#) (page 6-35)
- [Automatic CSV Encoding](#) (page 6-35)

 **See Also:**

["Specifying the Primary Language for an Application \(page 20-5\)"](#)

6.5.3.2.1 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.

6.5.3.2.2 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.

6.5.3.2.3 Document Direction

Sets the document direction. Options include:

- Left-To-Right
- Right-To-Left

6.5.3.2.4 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.
```

6.5.3.2.5 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.
```

6.5.3.2.6 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.SSXFF AM  
&MY_TIMESTAMP_FORMAT.
```

6.5.3.2.7 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.SSXFF AM TZR  
&MY_TIMESTAMP_TZ_FORMAT.
```

6.5.3.2.8 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
```

6.5.3.2.9 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`).

6.5.3.2.10 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

6.5.3.2.11 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.

6.5.4 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](#) (page 6-36)
- [User Interface Page](#) (page 6-37)
- [Defining an Application Logo](#) (page 6-39)
- [Creating a New User Interface](#) (page 6-40)
- [User Interface Details Page](#) (page 6-40)

6.5.4.1 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. Click **Shared Components**.
The Shared Components page appears.
4. Under User Interface, click **User Interface Attributes**.
The User Interface Attributes page appears. To learn more, see
5. To edit attributes for a specific user interface, in User Interfaces region and select a previously defined user interface.
The User Interface Details page appears.

6. Edit the appropriate attributes and click **Apply Changes**.

 **Tip:**

You can also access to User Interface attributes by navigating to the Edit Application Definition and then clicking the User Interface tab. See "[Accessing the Edit Application Definition Page](#) (page 6-16)."

 **See Also:**

"[User Interface Page](#) (page 6-37)" and "[User Interface Details Page](#) (page 6-40)"

6.5.4.2 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.

- [General Properties](#) (page 6-37)
- [Logo](#) (page 6-38)
- [Favicon](#) (page 6-39)
- [User Interface Detection](#) (page 6-39)
- [User Interfaces](#) (page 6-39)

 **See Also:**

"[Accessing User Interface Attributes](#) (page 6-36)"

6.5.4.2.1 General Properties

Use General Properties to define basic characteristics of the application user interface. [Table 6-13](#) (page 6-38) describes all General Properties attributes.

Table 6-13 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 (page 2-40)"</p>
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 text/html is used.</p>

6.5.4.2.2 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.



See Also:

"Defining an Application Logo (page 6-39)"

6.5.4.2.3 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 `#FAVICONS#` 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">
```

6.5.4.2.4 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.

6.5.4.2.5 User Interfaces

Displays user interfaces defined for the current application. To edit an existing user interface, click the user interface name. To add a new user interface, click **Add New User Interface**.



See Also:

["Accessing User Interface Attributes \(page 6-36\)"](#), ["Creating a New User Interface \(page 6-40\)"](#), and ["User Interface Details Page \(page 6-40\)"](#)

6.5.4.3 Defining an Application Logo

To define an application logo:

1. Access the User Interface page.
Next, specify the logo attributes.
2. 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.
3. 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.
4. 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](#) (page 16-14)" and "[About Referencing Static Workspace Files](#) (page 16-18)."

 **See Also:**

- "[Accessing User Interface Attributes](#) (page 6-36)"
- "[Managing Static Application Files](#) (page 16-12)"
- "[Managing Static Workspace Files](#) (page 16-17)"
- "[Creating Custom Themes](#) (page 11-32)"
- "[Page Templates](#) (page 11-59)"

6.5.4.4 Creating a New User Interface

To add a new user interface:

1. Navigate to the User Interface page.
2. In the User Interfaces region, click **Add New User Interface**.
3. Follow the on-screen instructions.
To learn more about an attribute, see field-level Help.
4. Confirm your selections and click **Create**.

 **See Also:**

"[Accessing User Interface Attributes](#) (page 6-36)" and "[User Interface Details Page](#) (page 6-40)"

6.5.4.5 User Interface Details Page

The User Interface Details page is divided into the following sections:

- [Identification](#) (page 6-41)
- [Attributes](#) (page 6-41)

- [Navigation Menu](#) (page 6-42)
- [Navigation Bar](#) (page 6-42)
- [JavaScript](#) (page 6-43)
- [Cascading Style Sheets](#) (page 6-44)
- [Concatenated Files](#) (page 6-44)



See Also:

"[Accessing User Interface Attributes](#) (page 6-36)"

6.5.4.5.1 Identification

Table 6-14 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.

6.5.4.5.2 Attributes

Table 6-15 User Interface Details, Attributes

Attribute	Description
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.
Default	Select whether the user interface is the default interface for the application.
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 Also: " Enabling Users to Select a Theme Style (page 11-5)"
Home URL	Specify the home page of the application for the current user interface.
Login URL	Specify the login page of the application for the current user interface.
Theme	Shows the theme currently associated with the user interface. See Also: " Switching Themes (page 11-21)"

Table 6-15 (Cont.) User Interface Details, Attributes

Attribute	Description
Theme Style	Select a theme style. This option only displays for newer themes that support theme styles. See Also: "Using Theme Styles (page 11-3)"
Global Page	If defined, displays the global page for the application. See Also: "Creating a Global Page to Display Components on Every Page (page 12-3)"

6.5.4.5.3 Navigation Menu

Navigation Menu attributes only display with newer themes.

Table 6-16 User Interface Details, Navigation Menu

Attribute	Description
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 on this application. <ul style="list-style-type: none"> Side renders the navigation menu list template in the <code>#SIDE_GLOBAL_NAVIGATION_LIST#</code> position on your page template. Top renders the navigation menu list template in the <code>#TOP_GLOBAL_NAVIGATION_LIST#</code> 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.

6.5.4.5.4 Navigation Bar

Navigation Bar attributes only display with newer themes.

Table 6-17 User Interface Details, Navigation Bar

Attribute	Description
Implementation	Select how you wish to implement the navigation bar in this application. <ul style="list-style-type: none"> Classic renders the navigation bar as a classic navigation bar in the <code>#NAVIGATION_BAR#</code> position on your page template. List renders the navigation bar as a list, using the selected list and list template in the <code>#NAVIGATION_BAR#</code> position on your page template.

Table 6-17 (Cont.) User Interface Details, Navigation Bar

Attribute	Description
Navigation Bar List	Select the List Template used to render 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.

6.5.4.5.5 JavaScript

Use these attributes to control or modify how an application handles JavaScript.

Table 6-18 User Interface Details, JavaScript

Attribute	Description
Content Delivery Network	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. 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.
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 <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. JavaScript file URLs you enter here replaces the <code>#APPLICATION_JAVASCRIPT#</code> substitution string in the page template. Note: You do not need to include opening or closing script tags. Just write the URL. Examples: <ul style="list-style-type: none"> Standard file reference: <pre>/myjs/main.js</pre> Standard file reference which loads the minified file <code>main.min.js</code> for regular page views and <code>main.js</code> in Debug mode: <pre>/myjs/main#MIN#.js</pre> Conditional file for Internet Explorer <pre>[if IE]/myjs/ie.js</pre>

Table 6-18 (Cont.) User Interface Details, JavaScript

Attribute	Description
Include Legacy JavaScript	<p>Specifies if the legacy JavaScript functions are included on every page in the application. If you are confident your application does not contain any references to the legacy functions, set this to No to reduce the overall size of the JavaScript files loaded.</p> <p>To determine what if any deprecated jQuery features are used, run the application with the Browser console log open and look for log messages displayed by jQuery Migrate.</p> <p>See Also: "Legacy JavaScript APIs" in <i>Oracle Application Express API Reference</i></p>
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>

6.5.4.5.6 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.

6.5.4.5.7 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.

6.6 Managing Pages in a Database Application

A page is the basic building block of an application. When you build an application in App Builder, you create pages that contain user interface elements, such as tabs, lists, buttons, items, and regions.

 **Tip:**

You view and manage application pages in App Builder using either Page Designer or Component view. To learn more, see "[Managing Pages in Page Designer](#) (page 7-1)" or "[Managing Application Legacy Components](#) (page 24-1)."

- [About Adding Pages to a Database Application](#) (page 6-45)
- [Page Types When Running the Create Page Wizard](#) (page 6-45)
- [Creating a Page from the Application Home Page](#) (page 6-48)
- [Creating Dialog Pages](#) (page 6-49)
- [About Copying Database Application Pages](#) (page 6-54)
- [About Copying Pages Between User Interfaces](#) (page 6-54)
- [About Deleting Database Application Pages](#) (page 6-55)

 **See Also:**

"[Using Application Utilities](#) (page 6-74)"

6.6.1 About Adding Pages to a Database Application

Developers add pages to an existing database application by running the Create Page Wizard. You can access this wizard as follows:

- On Application home page — Click **Create Page**. See "[Creating a Page from the Application Home Page](#) (page 6-48)."
- Viewing a page in Page Designer — Click **Create** on the Page Designer Developer toolbar and select **Page**. See "[Page Designer Toolbar](#) (page 7-3)"
- Viewing a page in Component View — Click **Create** on the Component View navigation bar and select **Page**. See "[Component View Navigation Bar](#) (page 24-5)."

6.6.2 Page Types When Running the Create Page Wizard

When you create an application, you choose a target user interface. The user interface you select determines the available page types. Not all page types are supported by all user interfaces.

Available Page Types for Desktop Applications

If you are creating a desktop application, the Create Page Wizard supports the following page types.

Table 6-19 Create Page Wizard — Page Types for Desktop Applications

Page Type	Description	To Learn More
Blank Page	Creates a page with no built-in functionality.	Not applicable
Report	A report is the formatted result of a SQL query. Report options: <ul style="list-style-type: none"> Interactive Report Interactive Grid Classic Report Report and Form on Table Report on Web Service 	See: <ul style="list-style-type: none"> "Developing Reports (page 8-1)" "Understanding Report Types (page 8-1)" "Report Options When Running the Create Page Wizard (page 8-7)"
Form	Forms enable users to update a single row or multiple rows in a table. Form options: <ul style="list-style-type: none"> Report with Form on Table Editable Interactive Grid Single Page Master Detail Two Page Master Detail 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 (page 9-5)" "Understanding Form Types (page 9-1)" "Form Options When Running the Create Page Wizard (page 9-6)"
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 (page 18-11)"
Chart	Enables you to create graphical charts.	See "Creating Charts (page 10-23)"
Calendar	Generates a calendar with monthly, weekly, and daily views.	See "Creating Calendars (page 10-1)"
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 "Creating Trees (page 13-61)"
Wizard	Creates a wizard.	Not applicable
Data Loading	Creates a new 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 (page 17-1)"
Feedback Page	Adds a feedback page. Feedback is the process of gathering real-time comments, enhancement requests, and bugs from your application users.	See "Managing Feedback (page 4-20)"

Table 6-19 (Cont.) Create Page Wizard — Page Types for Desktop Applications

Page Type	Description	To Learn More
Access Control	Creates a page containing an access control list, enabling developers to control access to an application, individual pages, or page components.	See " Controlling Access to Applications, Pages, and Page Components (page 19-39)"
Global Page	A global page (previously referred to as Page 0) functions as a master page.	See " Creating a Global Page to Display Components on Every Page (page 12-3)"
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 (page 24-1)"

Available Page Types for Mobile Applications

If you are creating a mobile application, the Create Page Wizard supports the following page types.

Table 6-20 Create Page Wizard — Page Types for Mobile Applications

Page Type	Description	To Learn More
Blank Page	Creates a page with no built-in functionality.	Not applicable
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 (page 8-1)" • "Understanding Report Types (page 8-1)" • "Report Options When Running the Create Page Wizard (page 8-7)"
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 (page 9-5)" • "Understanding Form Types (page 9-1)" • "Form Options When Running the Create Page Wizard (page 9-6)"
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 (page 18-11)"

Table 6-20 (Cont.) Create Page Wizard — Page Types for Mobile Applications

Page Type	Description	To Learn More
Chart	Enables you to create graphical charts.	See " Managing Legacy Charts (page 24-32)"
Calendar	Generates a calendar with monthly, weekly, and daily views.	See " Creating Calendars (page 10-1)"
Wizard	Create a wizard.	Not applicable
Feedback Page	Adds a feedback page. Feedback is the process of gathering real-time comments, enhancement requests, and bugs from your application users.	See " Managing Feedback (page 4-20)"
Access Control	Creates a page containing an access control list, enabling developers to control access to an application, individual pages, or page components.	See " Controlling Access to Applications, Pages, and Page Components (page 19-39)"
Legacy Page	Legacy pages contains components Oracle still supports but does not recommend usings. Options include: <ul style="list-style-type: none">• AnyChart Chart• Summary Page• Legacy Calendar	See " Managing Application Legacy Components (page 24-1)"

6.6.3 Creating a Page from the Application Home Page

To create a page from the Application home 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. For Select a page type, select the type of page you want to create. See "[Page Types When Running the Create Page Wizard](#) (page 6-45)."
 - c. Click **Next**.
5. Follow the on-screen instructions.

 **See Also:**

- ["About Adding Pages to a Database Application \(page 6-45\)"](#)
- ["Managing Page Attributes in Page Designer \(page 7-31\)"](#)
- ["Configuring Dialog Attributes in Component View \(page 24-27\)"](#)

6.6.4 Creating Dialog Pages

Oracle Application Express includes support for 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 \(page 6-49\)](#)
- [Using a Dialog Page Template \(page 6-49\)](#)
- [Defining a Page as a Dialog in Component View \(page 6-50\)](#)
- [Defining a Page as a Dialog in Page Designer \(page 6-51\)](#)
- [Generating a Cancel Dialog Dynamic Action on the Dialog Page \(page 6-52\)](#)
- [Generating a Close Dialog Page Process on the Dialog Page \(page 6-52\)](#)
- [About Branching with Dialog Pages \(page 6-53\)](#)
- [Handling Refresh Actions on Parent Page \(page 6-53\)](#)

6.6.4.1 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.

6.6.4.2 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 \(page 6-50\)](#)
- [About Defining Dialog Page Template Attributes \(page 6-50\)](#)

6.6.4.2.1 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](#) (page 11-34)."

6.6.4.2.2 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](#) (page 11-67)"

6.6.4.3 Defining a Page as a Dialog in Component View

To create a dialog page you must set the Page Mode to either **Modal Dialog** or **Non-modal Dialog**. You can configure the page mode when running the Create Page Wizard or by editing the page attributes.

To define a page as a dialog in Component View:

1. Edit the page attributes. View the page in Component View.

2. Set Page Mode to **Modal Dialog** and **Non-Modal Dialog**.

 **Tip:**

You can configure the Page Mode by editing the page attributes or when running the Create Page Wizard.

3. Define the page-level Dialog attributes.

 **See Also:**

["Editing Page Attributes in Component View \(page 24-13\)"](#) and ["Dialog \(page 24-17\)"](#)

6.6.4.4 Defining a Page as a Dialog 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 Adding Pages to a Database Application \(page 6-45\)"](#)

6.6.4.5 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 \(page 13-36\)"](#)

6.6.4.6 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 \(page 15-22\)"](#)
- ["Understanding Sample Database Application \(page 3-8\)"](#)
- ["Controlling Navigation Using Branches \(page 14-25\)"](#)

6.6.4.7 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.

6.6.4.8 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 \(page 3-8\)"](#)
- ["Managing Dynamic Actions \(page 13-36\)"](#)
- ["Understanding Sample Database Application \(page 3-8\)"](#)

6.6.5 About Copying Database Application Pages

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.

 **See Also:**

["Copying a Database Application Page in Page Designer \(page 7-39\)"](#) and ["Copying a Database Application Page in Legacy Component View \(page 24-30\)"](#)

6.6.6 About Copying Pages Between User Interfaces

 **Note:**

Oracle does not support copying Desktop pages to jQuery Mobile Smartphone pages or copying jQuery Mobile Smartphone pages to Desktop pages.

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.

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

6.6.7 About Deleting Database Application Pages

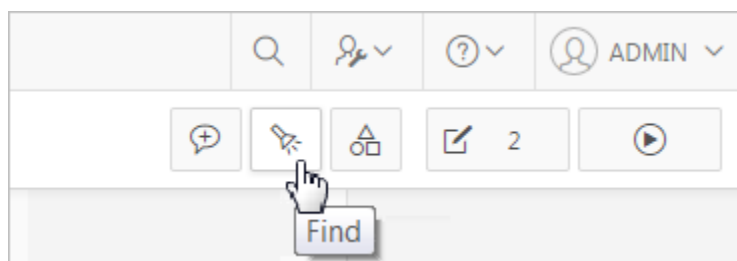
Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries. You can only delete unlocked pages or pages that you have locked.

See Also:

["Deleting Pages in Page Designer \(page 7-40\)"](#) and ["Deleting Pages in Legacy Component View \(page 24-31\)"](#)

6.7 Using the Find Icon

The Find icon resembles a flashlight and displays on many pages in App Builder including Application home page, Legacy Component View, Shared Components, and the Edit Application Definition.



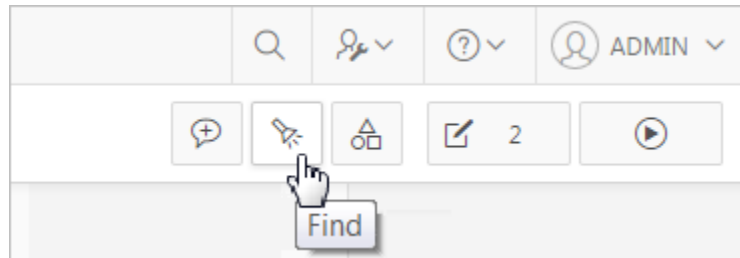
Clicking the Find icon displays a dialog that enables you to search for items, pages, queries, tables, PL/SQL code, or images, view debug reports, view session state, and view errors.

- [Using the Find Icon \(page 6-56\)](#)
- [Searching for Items \(page 6-56\)](#)
- [Searching for Pages \(page 6-57\)](#)
- [Searching for Queries \(page 6-58\)](#)
- [Searching for Tables \(page 6-59\)](#)
- [Using the PL/SQL Finder \(page 6-59\)](#)
- [Searching for Images \(page 6-60\)](#)

6.7.1 Using the Find Icon

To use the Find icon:

1. Click the **Find** icon.



The Items Finder appears.

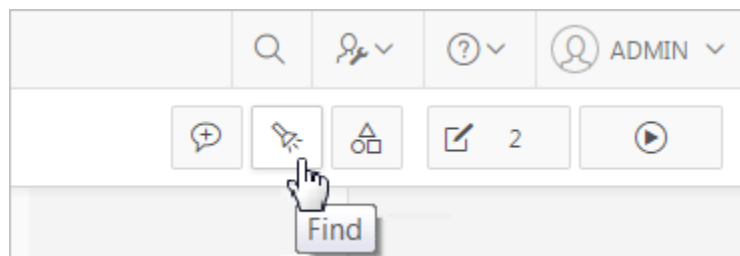
2. Click one of the following tabs:
 - Items - See "[Searching for Items](#) (page 6-56)."
 - Pages - See "[Searching for Pages](#) (page 6-57)."
 - Queries - See "[Searching for Queries](#) (page 6-58)."
 - Tables - See "[Searching for Tables](#) (page 6-59)."
 - PL/SQL - See "[Using the PL/SQL Finder](#) (page 6-59)."
 - Images - See "[Searching for Images](#) (page 6-60)."
 - Debug - See "[Viewing Debug Reports](#) (page 22-7)."
 - Session - See "[Understanding Session State Management](#) (page 2-13)" and "[Using the Session Page](#) (page 2-15)."
 - Errors - View errors.

6.7.2 Searching for Items

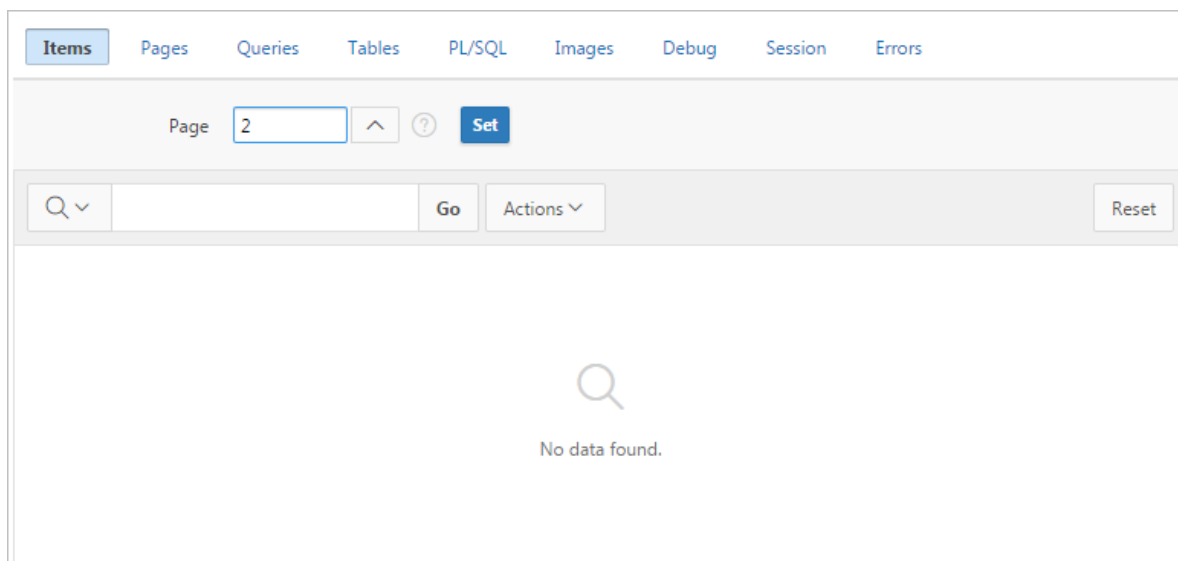
In App Builder, an item can be a text field, text area, password, select list, check box, and so on. You can use the Item Finder to search for items within the current application or within the schema associated with the workspace.

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:

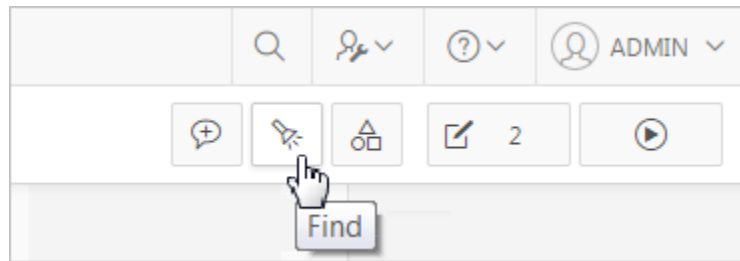
["Understanding Page-Level Items \(page 13-1\)"](#)

6.7.3 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.

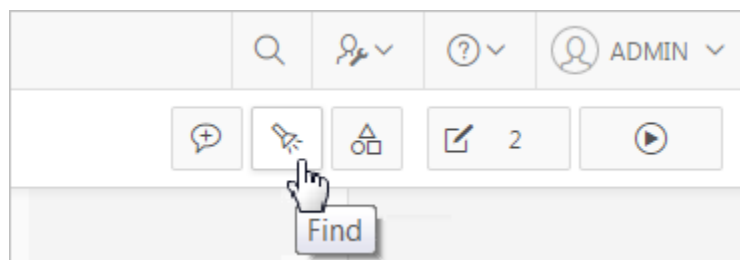
3. To link to the Items page, click the page number.

6.7.4 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.

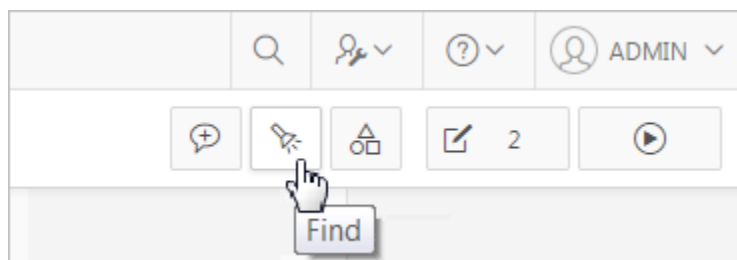
- **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report. See "[About the Actions Menu](#) (page 8-94)."
3. To link to the Items page, click the page number.

6.7.5 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:

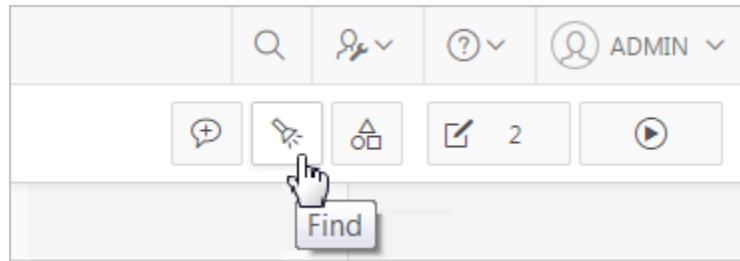
"Managing Tables" in *Oracle Application Express SQL Workshop Guide*

6.7.6 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:

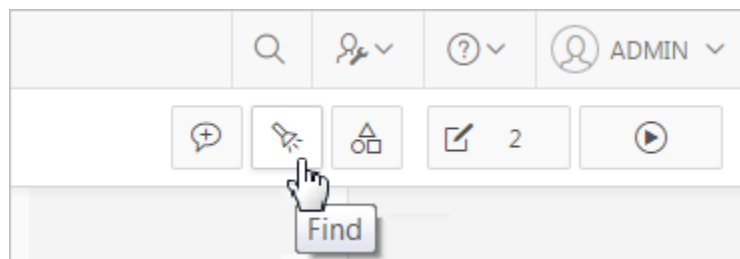
"Managing Database Objects with Object Browser" in *Oracle Application Express SQL Workshop Guide*

6.7.7 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*

6.8 Creating Page Groups

You can make the pages within your application easier to access by organizing them into page groups. 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](#) (page 6-61)
- [Creating a Page Group](#) (page 6-62)
- [Editing a Page Group Definition](#) (page 6-62)
- [Assigning a Page to a Page Group](#) (page 6-63)
- [Viewing Pages by Page Group](#) (page 6-63)
- [Deleting a Page Group](#) (page 6-63)

6.8.1 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 \(page 8-94\)"](#)

6.8.2 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**.

6.8.3 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.

6.8.4 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**.

6.8.5 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.

6.8.6 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.

6.9 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](#) (page 6-64)
- [Locking a Page](#) (page 6-64)
- [Unlocking a Page](#) (page 6-66)

6.9.1 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
1	Home	3 weeks ago	-	Home	Unassigned	Desktop		
2	Interactive Report	67 seconds ago	admin_user	Interactive Report	Unassigned	Desktop		
5	Access Control Administration Page	23 hours ago	admin_user	DML Form	Unassigned	Desktop		
101	Login Page	3 weeks ago	-	Login	Unassigned	Desktop		

6.9.2 Locking a Page

Developers can lock a page on the Application home page, Page Locks page, in Page Designer, and in Component View.

- [Locking a Page from the Application Home Page](#) (page 6-65)
- [Locking a Page from the Page Locks Page](#) (page 6-65)
- [Locking a Page in Page Designer](#) (page 6-65)
- [Locking a Page in Legacy Component View](#) (page 6-66)

6.9.2.1 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)**.

6.9.2.2 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)**.

6.9.2.3 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 \(page 7-22\)"](#)

6.9.2.4 Locking a Page in Legacy Component View

To lock a page in Component View:

1. Navigate to the appropriate page in Component View.
2. Click **Utilities** and then **Lock**.
3. Enter a comment in the Comment field.
4. Click **Lock Page(s)**.



See Also:

["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

6.9.3 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 \(page 6-66\)](#)
- [Unlocking Pages from the Application Home Page \(page 6-67\)](#)
- [Unlocking a Page in Page Designer \(page 6-67\)](#)
- [Unlocking a Page in Legacy Component View \(page 6-67\)](#)
- [Unlocking Pages As a Workspace Administrator \(page 6-67\)](#)

6.9.3.1 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**.

6.9.3.2 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**.

6.9.3.3 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 \(page 7-22\)"](#)

6.9.3.4 Unlocking a Page in Legacy Component View

To unlock pages in Component View:

1. Navigate to the appropriate page in legacy Component View.
2. Click **Utilities** and select **Unlock**.



See Also:

See ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

6.9.3.5 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)**.

6.10 Adding Developer Comments

Developers can 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.

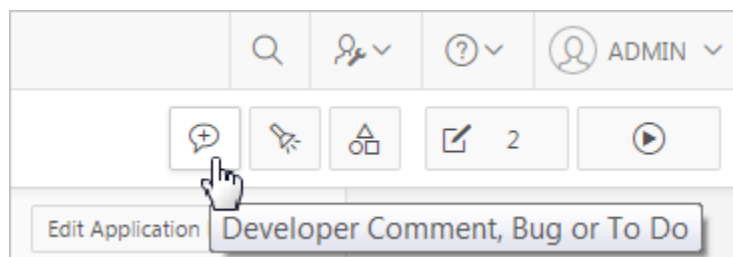
- [Developer Comment, Bug, or To Do Button](#) (page 6-68)
- [Adding Developer Comments to an Application](#) (page 6-69)
- [Viewing and Editing Developer Comments](#) (page 6-69)
- [Deleting Developer Comments](#) (page 6-69)
- [Accessing the Developer Comments from Shared Components](#) (page 6-70)
- [Viewing the Developer Comments Calendar](#) (page 6-71)

See Also:

["Adding Database Application Comments](#) (page 6-71)"

6.10.1 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.



6.10.2 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**.

6.10.3 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](#) (page 6-70)."

6.10.4 Deleting Developer Comments

You can delete specific developer comments or delete multiple comments at once.

- [Deleting a Specific Developer Comment](#) (page 6-69)
- [Deleting Multiple Developer Comments](#) (page 6-70)

6.10.4.1 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**.

6.10.4.2 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.

6.10.5 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 \(page 8-94\)."](#)



See Also:

["Viewing and Editing Developer Comments \(page 6-69\)"](#) and ["Deleting Developer Comments \(page 6-69\)"](#)

6.10.6 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.

6.11 Adding Database Application Comments

Developers can add comments concerning an entire database application on the Application Comments page. You could add a description of the application or track the developers involved in working on the application.

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**.

6.12 Managing Application Groups

Developers can 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 \(page 6-72\)](#)

- [Creating a Database Application Group](#) (page 6-72)
- [Assigning a Database Application to an Application Group](#) (page 6-72)
- [Removing a Database Application from an Application Group](#) (page 6-73)
- [Deleting an Database Application Group](#) (page 6-73)



See Also:

["Accessing Page Specific Utilities](#) (page 6-79)"

6.12.1 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.

6.12.2 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**.

6.12.3 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](#) (page 6-16)."

6.12.4 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](#) (page 6-16)."

6.12.5 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.

6.13 Using Application Utilities

The Utilities page features links to summaries and reports that enable developers to view an application from various perspectives.

- [Accessing the Utilities Page](#) (page 6-74)
- [Viewing the Application Dashboard](#) (page 6-75)
- [Upgrading an Application to Include New Components](#) (page 6-76)
- [Viewing the Recently Updated Pages Report](#) (page 6-77)
- [Viewing the Change History Report](#) (page 6-77)
- [Using the Database Object Dependencies Report](#) (page 6-78)
- [Accessing Application Express Views](#) (page 6-78)
- [Accessing Page Specific Utilities](#) (page 6-79)
- [Accessing Cross Page Utilities](#) (page 6-80)
- [Accessing Cross Application Reports](#) (page 6-80)
- [Using Interactive Report Utilities](#) (page 6-81)



See Also:

["Creating Custom Activity Reports Using APEX_ACTIVITY_LOG \(page 21-8\)"](#)

6.13.1 Accessing the Utilities Page

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.** View a summary of application components and attributes.

See "[Viewing the Application Dashboard](#) (page 6-75)."

- **Advisor.** Performs various checks on your application, including checks for programming errors, security issues, quality assurance, and other best practices.
See "[Running Advisor to Check Application Integrity](#) (page 22-1)."
- **Upgrade Application.** Upgrade the current application to a new release. Review components eligible for upgrading to include the latest features.
See "[Running Upgrade Application](#) (page 6-77)."
- **Recently Updated Pages.** View a history of page updates made to this application.
See "[Viewing the Recently Updated Pages Report](#) (page 6-77)."
- **Attribute Dictionary.** Manage item and column user interface defaults for a selected page.
See "[Using the Attribute Dictionary](#) (page 15-40)."
- **Change History.** View a report of component updates made in this application.
See "[Viewing the Change History Report](#) (page 6-77)."
- **Database Object Dependencies.** View a report of database object referenced by this application.
See "[Using the Database Object Dependencies Report](#) (page 6-78)."
- **Debug Messages.** Review debug messages generated by this application.
See "[Debugging an Application](#) (page 22-1)."
- **Application Express Views.** Query various views against Application Express metadata.
See "[Accessing Application Express Views](#) (page 6-78)."
- **Export Repository.** View a history of application exports.
See "[Accessing the Export Repository](#) (page 23-36)."

**See Also:**

"[Accessing Page Specific Utilities](#) (page 6-79)"

6.13.2 Viewing the Application Dashboard

The Application Dashboard details 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.
6. To view a specific report, click the number to the right of the component type or template.



See Also:

["Identifying At Risk Password Items \(page 19-19\)"](#)

6.13.3 Upgrading an Application to Include New Components

Developers can upgrade an existing application to include new components available in the current release.

- [About Upgrading to Include New Components \(page 6-76\)](#)
- [Running Upgrade Application \(page 6-77\)](#)

6.13.3.1 About Upgrading to Include New Components

Using **Upgrade Application** on the Utilities page upgrades the following components:

- Upgrade AnyChart charts to Oracle JET charts.
- Upgrade Yes and No radio groups and select lists to a switch.
- Upgrade tabular form to interactive grid.

The following attributes are not migrated:

- Default column sort sequence and direction
- When No Data Found message
- When When More Data Found message

- Maximum Row Count



See Also:

["Running Upgrade Application \(page 6-77\)"](#)

6.13.3.2 Running Upgrade Application

To upgrade an existing application:

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**.
6. Follow the on-screen instructions.



See Also:

["About Upgrading to Include New Components \(page 6-76\)"](#)

6.13.4 Viewing the Recently Updated Pages Report

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**.
A Search bar displays at the top of the page.

6.13.5 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**.

6.13.6 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.

6.13.7 Accessing Application Express Views

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 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 view by name and click **Go**.
- **Go button** - Executes a search or applies a filter.
- **View Icons** - Displays each view as a large icon. To access a view, click the appropriate icon.
- **View Report** - Displays each view as a line in a report. To access a view, click the name.
- **Actions menu** - Use the Actions menu to customize the report view. See "[About the Actions Menu](#) (page 8-94)."

6. Select a view

To view the data within a view, select columns from the Select Column tab and click the **Results** button.

On the Results tab, expand the Query region to view the query that was executed. You can copy this query for reuse with SQL Commands or SQL Developer.

6.13.8 Accessing Page Specific Utilities

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 \(page 6-80\)"](#)

6.13.9 Accessing Cross Page Utilities

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 \(page 6-64\)"](#)
- ["Deleting Multiple Pages \(page 24-31\)"](#)
- ["Using Application Utilities \(page 6-74\)"](#)

6.13.10 Accessing Cross Application Reports

Use the links on the Cross Application Reports page to view and edit information across multiple applications in the current workspace. You access cross application reports from Workspace Utilities page.

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 **Cross Application Reports** on the right side of the window.
5. Under, Cross Application Reports, click **All Reports** to view the Cross Application Reports page.
6. Select a report to view.

6.13.11 Using Interactive Report Utilities

This section describes options under Interactive Report Utilities on the right side of the Utilities page.

- [Deleting Saved Interactive Reports](#) (page 6-81)
- [Viewing Subscriptions](#) (page 6-82)

6.13.11.1 Deleting Saved Interactive Reports

The Saved Interactive Reports page lists saved interactive reports in the currently selected application.

To manage 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](#) (page 8-91)"
- ["Saving Interactive Reports](#) (page 8-96)"
- "Managing Saved Interactive Reports" in *Oracle Application Express Administration Guide*

6.13.11.2 Viewing 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 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 **Subscriptions**.
A report appears.

7

Managing Pages in Page Designer

Developers view and manage application in Page Designer. Page Designer is a full featured Integrated Development Environment (IDE) that includes a toolbar and multiple panes.

Topics:

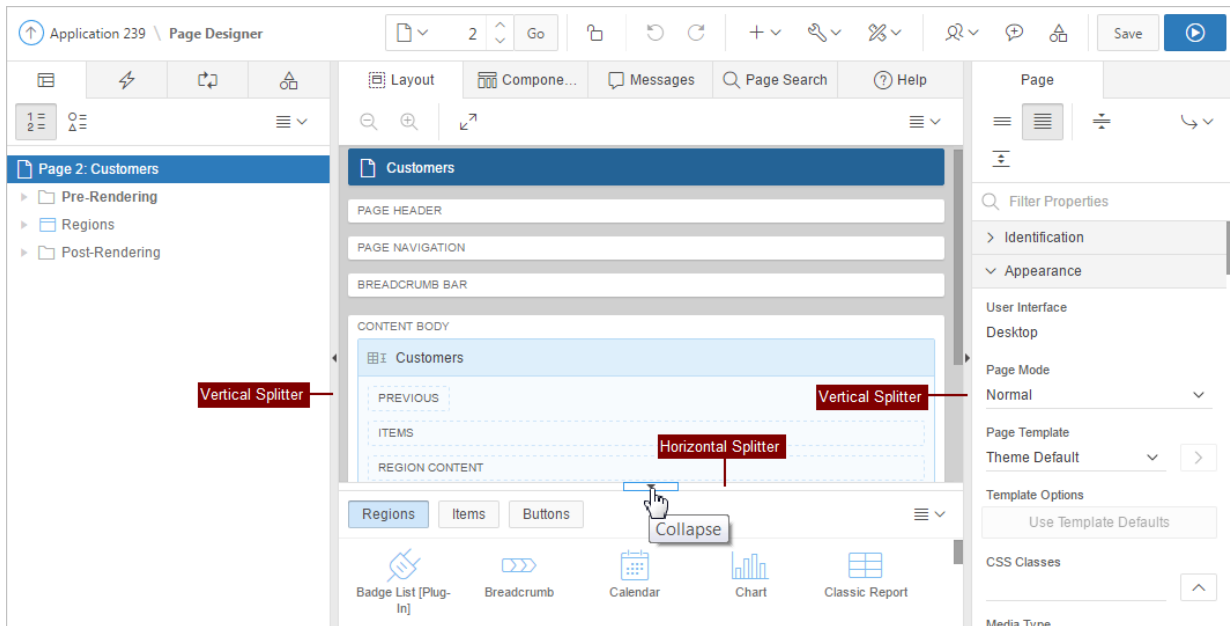
- [About Page Designer](#) (page 7-1)
- [Understanding Page Designer UI Elements](#) (page 7-3)
- [Using Page Designer](#) (page 7-21)
- [Managing Page Attributes in Page Designer](#) (page 7-31)
- [Running a Page from Page Designer](#) (page 7-37)
- [Runtime Developer Toolbar](#) (page 7-38)
- [Copying a Database Application Page in Page Designer](#) (page 7-39)
- [Deleting Pages in Page Designer](#) (page 7-40)
- [Switching from Legacy Component View to Page Designer](#) (page 7-41)

See Also:

- ["Adding Developer Comments](#) (page 6-68)"
- ["Creating Database Applications](#) (page 6-1)"
- ["Controlling Page Layout](#) (page 12-1)"
- ["Adding Navigation](#) (page 14-1)"

7.1 About Page Designer

Page Designer is a full featured Integrated Development Environment (IDE) that includes a toolbar and multiple panes. You can use Page Designer to maintain and enhance pages within an Oracle Application Express application.



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. See "[Left Pane of Page Designer](#) (page 7-7)."
- **Central Pane** - Includes five tabs: Layout, Component View, Messages, Page Search, and Help. See "[Central Pane of Page Designer](#) (page 7-11)."
- **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. See "[Right Pane of Page Designer \(Property Editor\)](#) (page 7-18)."

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:

"[Viewing Help in Page Designer](#) (page 1-28)," "[Managing Pages in Page Designer](#) (page 7-1)," "[Understanding Page Designer UI Elements](#) (page 7-3)," and "[Viewing a Page in Legacy Component View](#) (page 24-10)"

7.2 Understanding Page Designer UI Elements

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. Use the Property Editor to edit attributes of the selected component.

Tip:

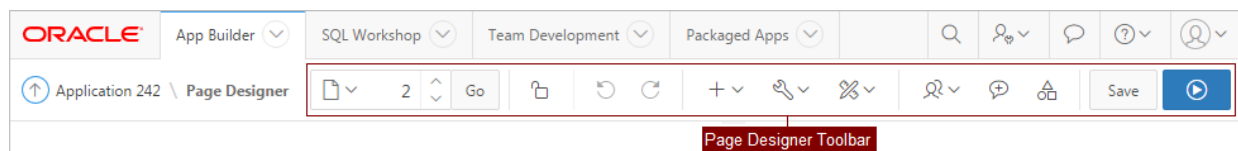
You can view an online summary of this information. View a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.

- [Page Designer Toolbar](#) (page 7-3)
- [Left Pane of Page Designer](#) (page 7-7)
- [Central Pane of Page Designer](#) (page 7-11)
- [Right Pane of Page Designer \(Property Editor\)](#) (page 7-18)
- [Code Editor](#) (page 7-20)

See Also:

["Using Page Designer](#) (page 7-21)"

7.2.1 Page Designer Toolbar



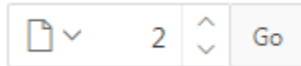
The Page Designer toolbar displays at the top of the page and contains both buttons and menu options. Passing your cursor over an active button and menu, displays a tooltip.

 **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 \(page 7-23\)."](#)

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 \(page 6-64\)."](#)

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 Adding Pages to a Database Application](#) (page 6-45)."
- **Page as Copy.** Access the Copy Page Wizard. See "[Copying a Database Application Page in Page Designer](#) (page 7-39)."
- **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 in Page Designer](#) (page 9-10)."
- **Breadcrumb Region.** Access to the Create Breadcrumb Wizard. See "[Creating Breadcrumbs](#) (page 14-8)" and "[Creating a Breadcrumb Region in Page Designer](#) (page 14-11)."
- **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 16-1)."
- **Page Group.** Links to the Page Group page. Use page groups to organize pages. See "[Creating Page Groups](#) (page 6-61)."
- **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](#) (page 6-68)."
- **Team Development.** Displays the submenus Feature, To Do, and Bug. See "[Tracking Features](#) (page 4-3)," "[Tracking To Dos](#) (page 4-13)," and "[Managing Bugs](#) (page 4-16)."

Utilities



Features a graphic of a wrench. Utilities menu options include:

- **Delete Page.** Delete the current page. See "[Deleting Pages in Page Designer](#) (page 7-40)."
- **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](#) (page 22-1)."
- **Caching.** Links to the Caching page. Enabling caching is an effective way to improve the performance. See "[Utilizing Region Caching in Page Designer](#) (page 12-9)" and "[Utilizing Region Caching in Component View](#) (page 12-22)."
- **Attribute Dictionary.** Access the Attribute Dictionary. See "[Using the Attribute Dictionary](#) (page 15-40)."
- **History.** Displays a report of changes to the current page.

- **Export.** Export the current page. See "[Exporting an Application and Application Components](#) (page 23-16)."
- **Cross Page Utilities.** Access Cross Page Utilities. See "[Accessing Cross Page Utilities](#) (page 6-80)."
- **Application Utilities.** Access Application Utilities. See "[Using Application Utilities](#) (page 6-74)"
- **Page Groups.** Links to the Page Group page. Use page groups to organize pages. See "[Creating Page Groups](#) (page 6-61)."
- **Upgrade Application.** Upgrade an existing application. See "[Running Upgrade Application](#) (page 6-77)."

Settings



Controls the number of panes that display in Page Designer. See "[Switching Between Three Pane and Two Pane Mode](#) (page 7-23)."

Team Development



Features a graphical outline of two people. Team Development menu options include:

- **Features.** Links to Create Feature. See "[Tracking Features](#) (page 4-3)."
- **To Do -** Links to Create To Do. See "[Tracking To Dos](#) (page 4-13)."
- **Bugs -** Links Create Bug. See "[Managing Bugs](#) (page 4-16)."
- **Feedback Entries -** Links to the Feedback report. See "[Managing Feedback](#) (page 4-20)."

Tip:

Team Development only displays if this feature is enabled for the workspace. See "Managing Team Development" in *Oracle Application Express Administration Guide*.

Developer Comments



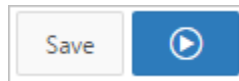
Displays the Create a Developer Comment, Bug, or To Do dialog. See "[Adding Developer Comments](#) (page 6-68)."

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](#) (page 16-1)."

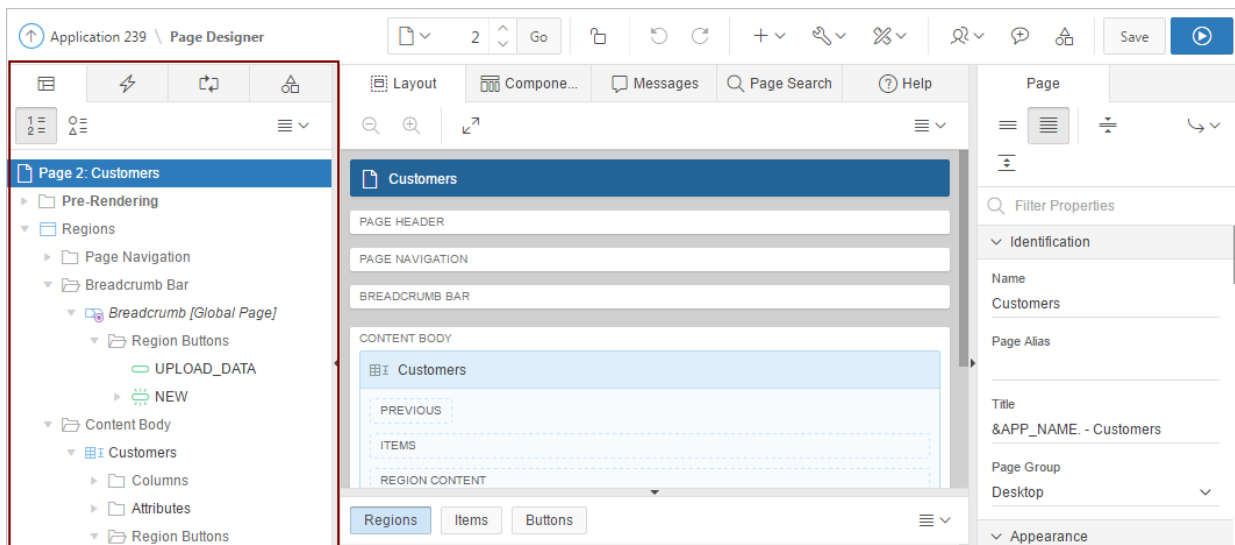
Save and Save and Run Page



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](#) (page 7-37)."

7.2.2 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:

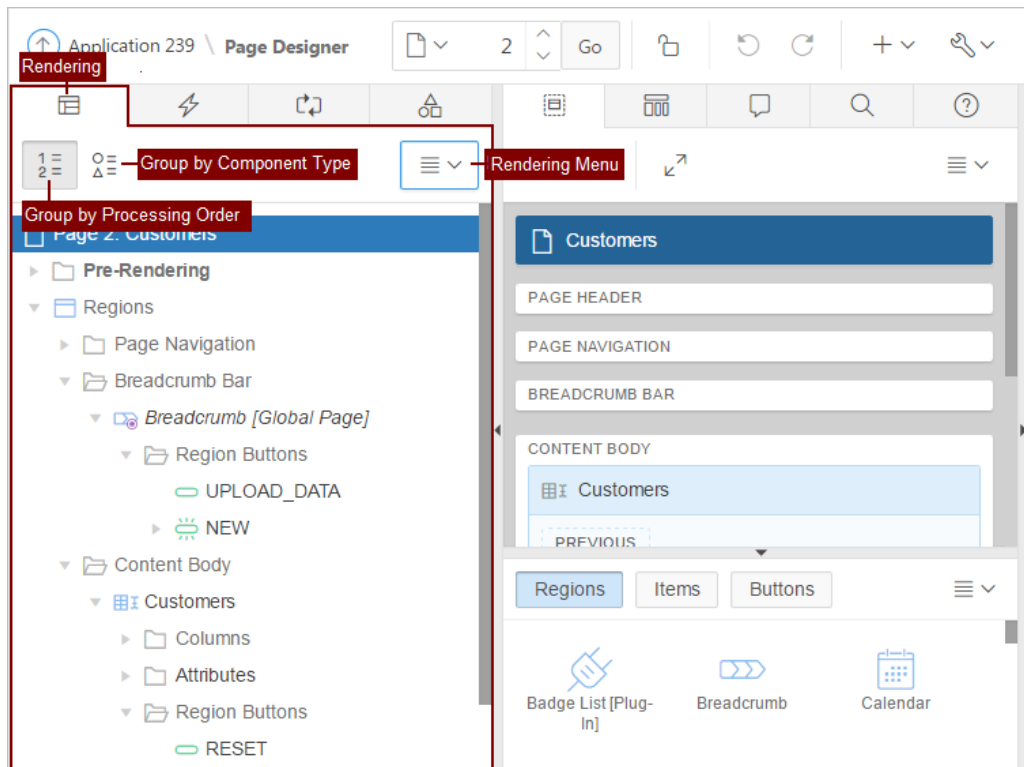
- **Context menus.** Right-click a component or control to display a context menu.
- **Quick Access to the Property Editor.** Select a component or multiple components to display the corresponding attributes in the Property Editor in the right pane.
- **Drag and Drop.** Drag components up and down within 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.

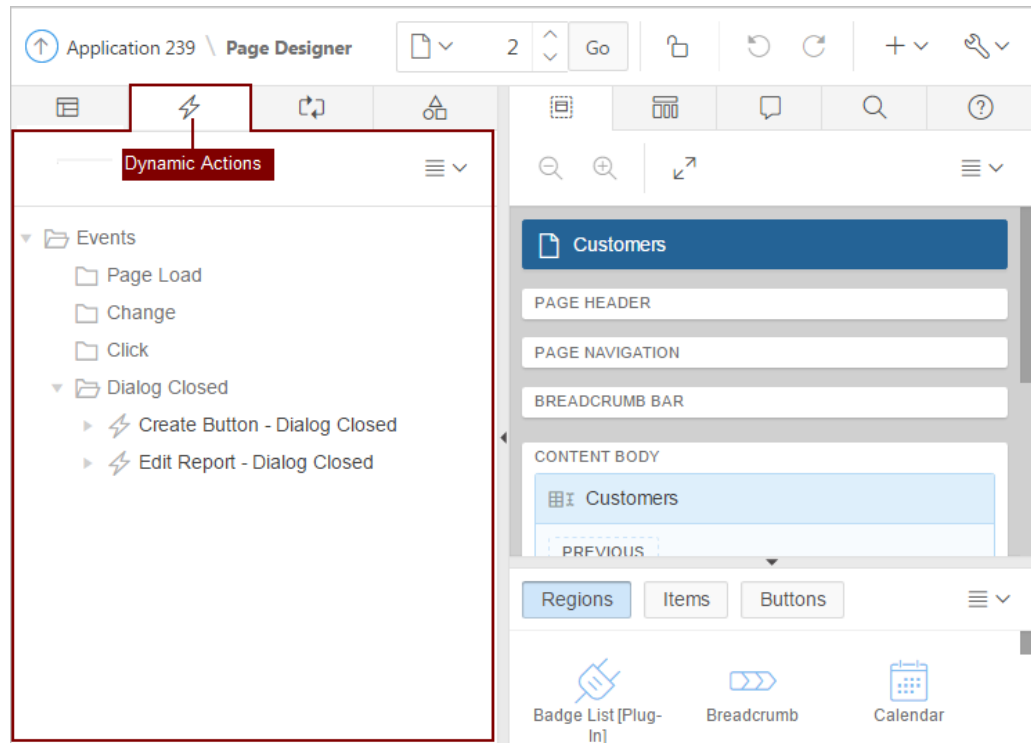
You can alter the display by clicking the following icons:

- **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.

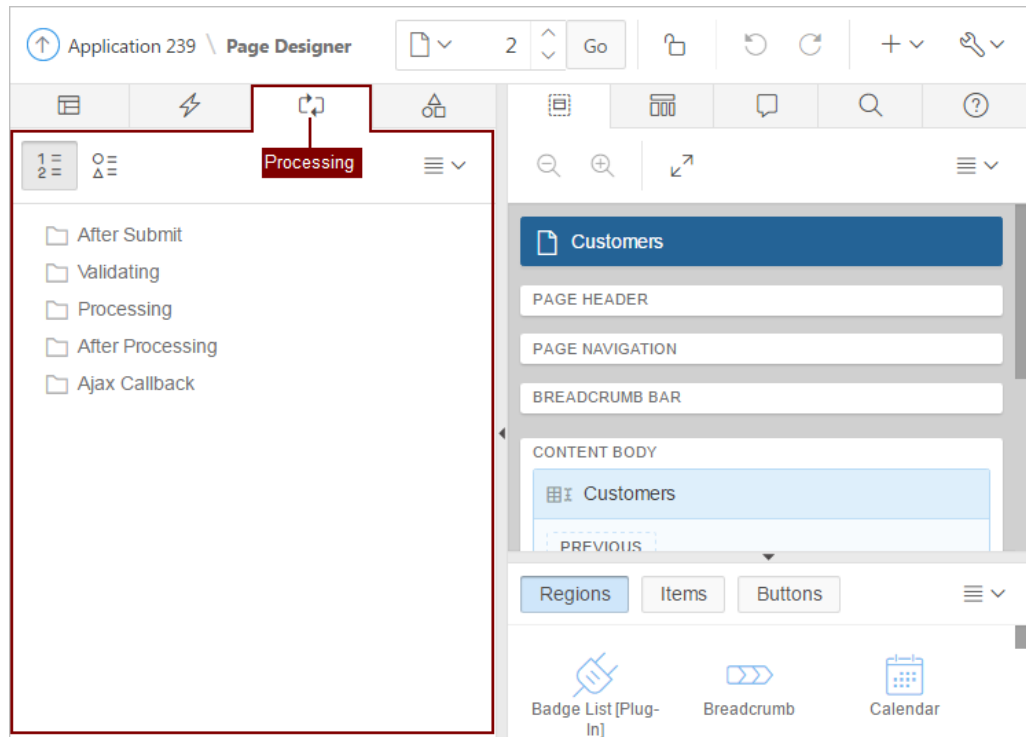


See Also:

["Managing Dynamic Actions \(page 13-36\)"](#)

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**

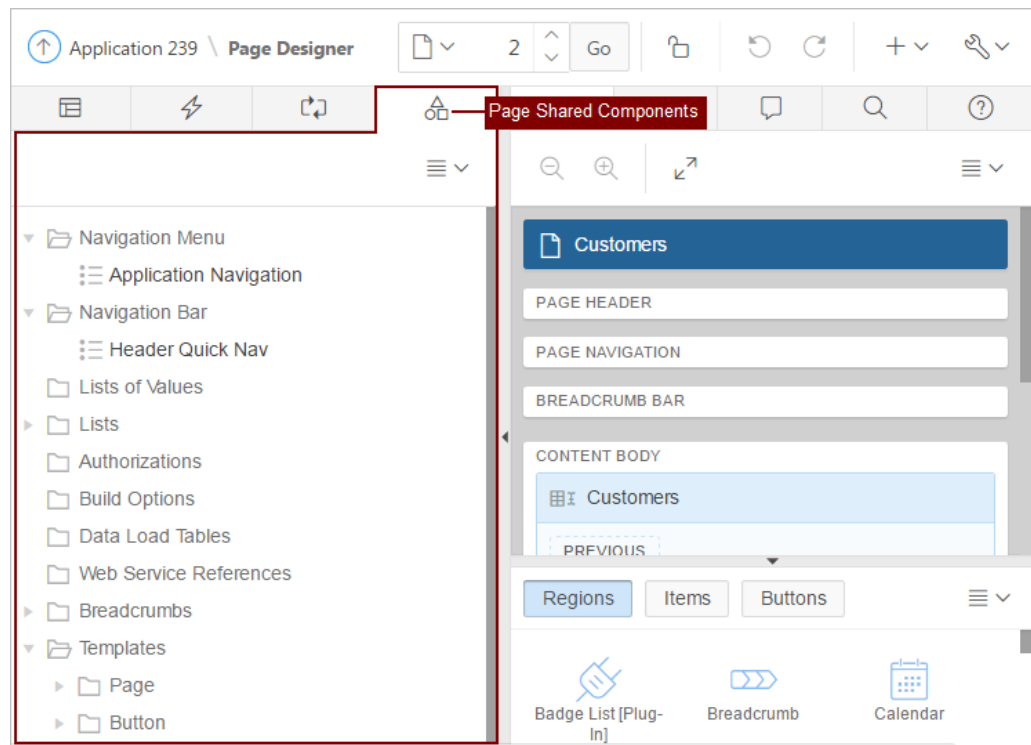


See Also:

["Managing Computations, Validations, and Processes \(page 15-1\)"](#)

Page Shared Components Tab

The Page Shared Components tab displays shared components associated with this page.



 **See Also:**

["Managing Shared Components \(page 16-1\)"](#)

7.2.3 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.

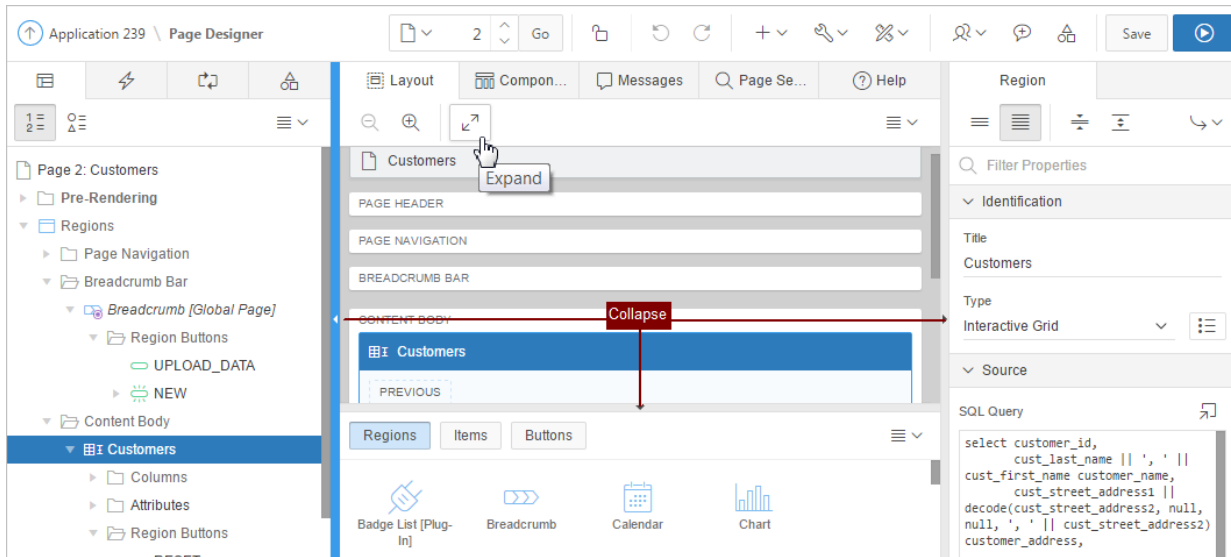
- [Layout Tab](#) (page 7-11)
- [Gallery](#) (page 7-13)
- [Component View Tab](#) (page 7-14)
- [Messages Tab](#) (page 7-15)
- [Page Search Tab](#) (page 7-16)
- [Help Tab](#) (page 7-17)

7.2.3.1 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:

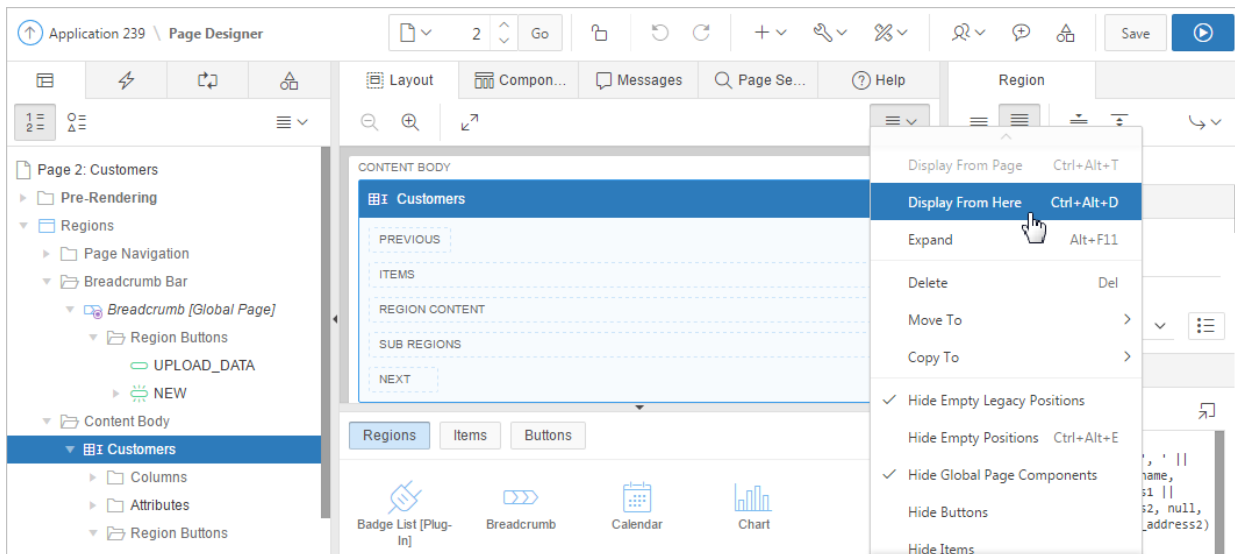
- **Context menus.** Right-click a component or control to display a context menu.
- **Quick Access to the Property Editor.** Select a component or multiple components to display the corresponding attributes in the Property Editor in the right pane.
- **Drag and Drop .** Move, copy, and reorder regions, 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](#) (page 7-25)".

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**.

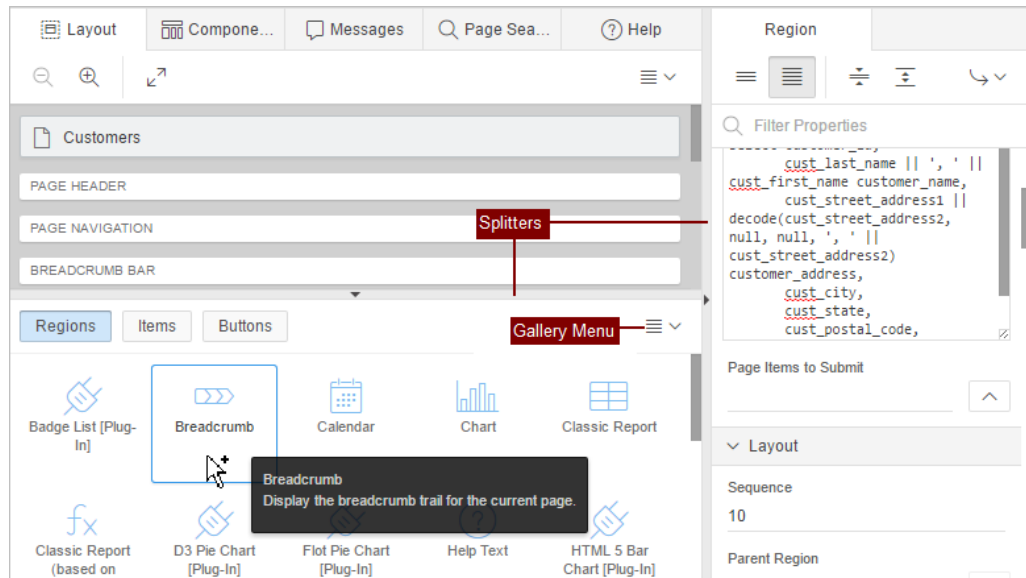


See Also:

"[Adding a Region, Item, or Button by Dragging and Dropping](#) (page 7-25)"

7.2.3.2 Gallery

The Gallery pane displays at the bottom of the central pane of Page Designer and 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](#) (page 7-25)"
and "[About Adding Components Using Gallery Context Menus](#) (page 7-26)"

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](#) (page 24-1)"

7.2.3.3 Component View Tab

Similar to the Legacy Component View window in **App Builder**, the Component View tab groups user interface elements and application logic by component type. The Component View tab displays in the central pane of Page Designer and 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.
- **Shared Components.** The Shared Components section lists common components that can be used by one or more pages within an application.

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.

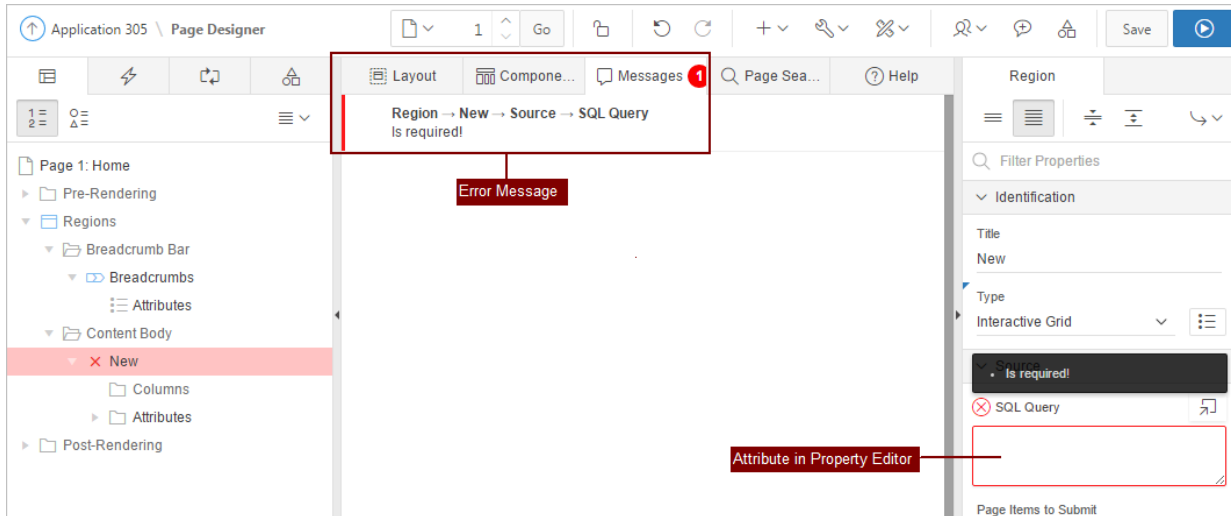
To add a new control, component, or process, click the **Create** icon to the right of the section name.

 **See Also:**

"[Adding a Region, Item, or Button from the Component View Tab](#) (page 7-28)" and "[Managing Pages in Legacy Component View](#) (page 24-1)"

7.2.3.4 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:**

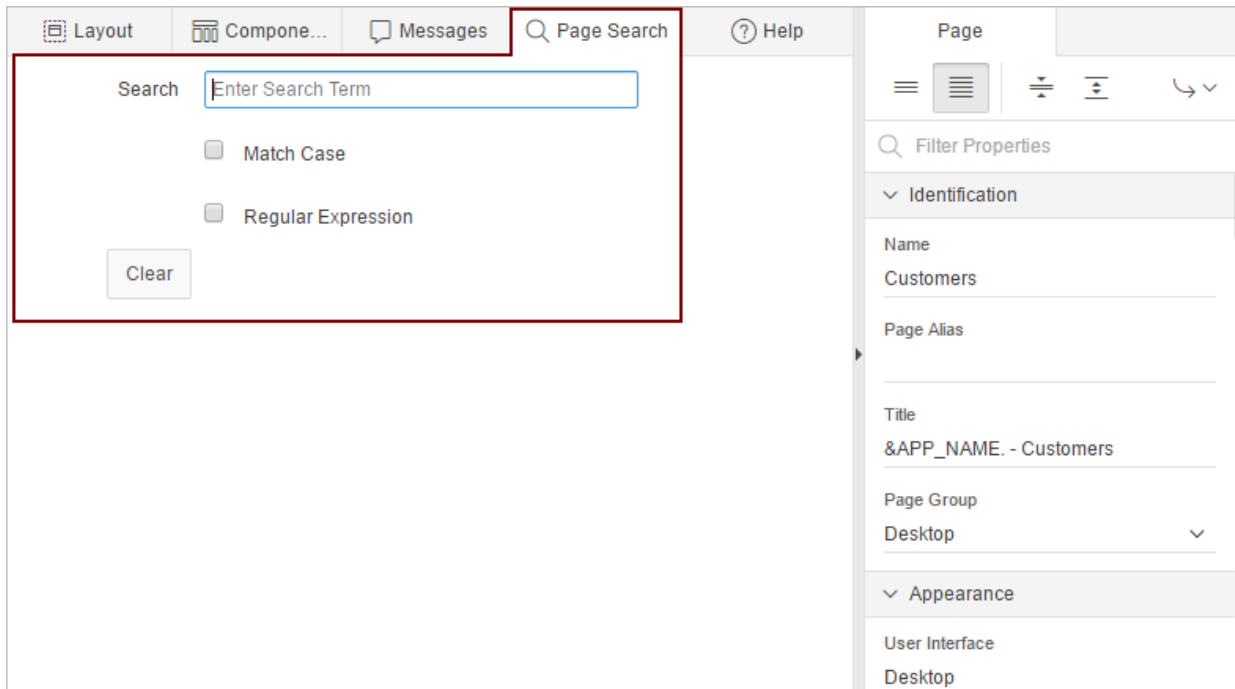
Components that have errors or warnings display in red with a X icon or yellow with a triangle icon in both the left pane and the Layout tab.

 **See Also:**

["Adding a Region, Item, or Button by Dragging and Dropping \(page 7-25\)"](#)
and ["About Adding Components Using Gallery Context Menus \(page 7-26\)"](#)

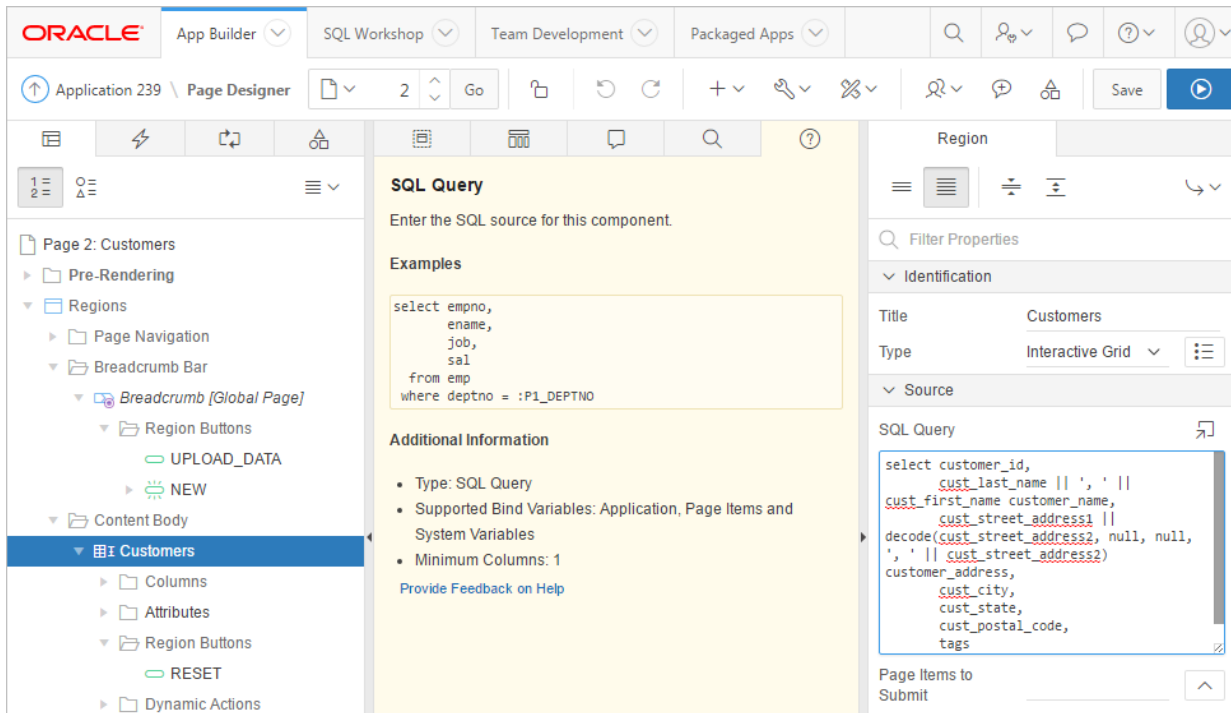
7.2.3.5 Page Search Tab

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**.



7.2.3.6 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 SQL Query for the Customers region is selected and the central pane displays the Help for the SQL Query attribute. Additional information about this illustration can be found in the surrounding text.



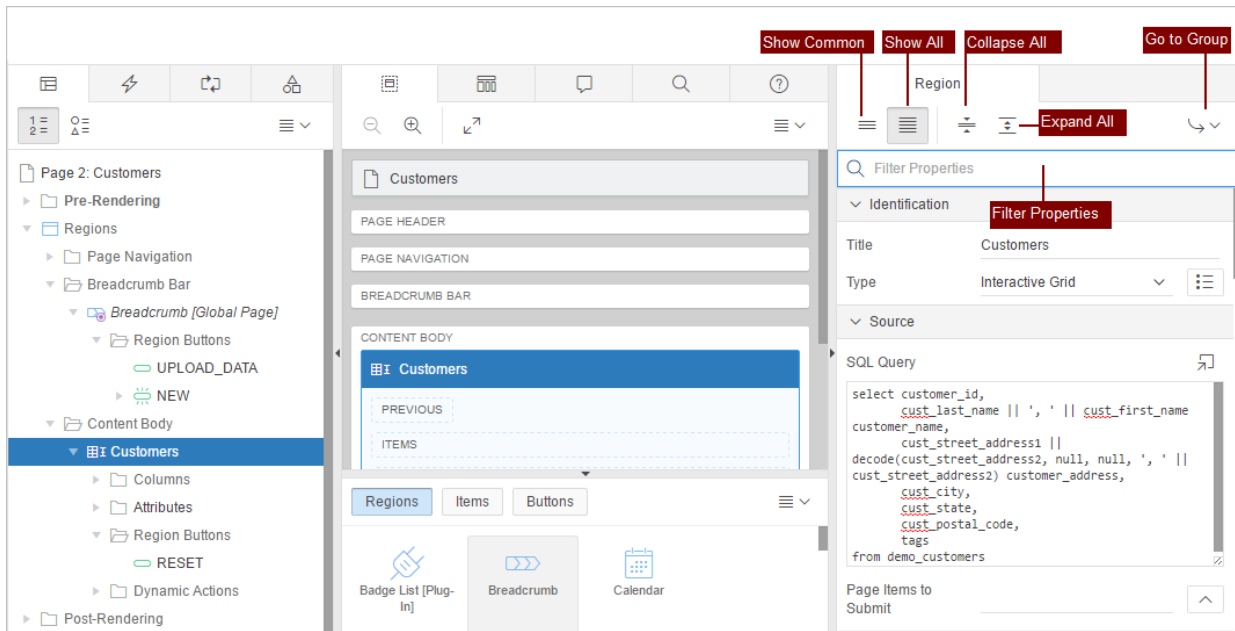
See Also:

["Viewing Help in Page Designer \(page 1-28\)"](#)

7.2.4 Right Pane of Page Designer (Property Editor)

Property Editor

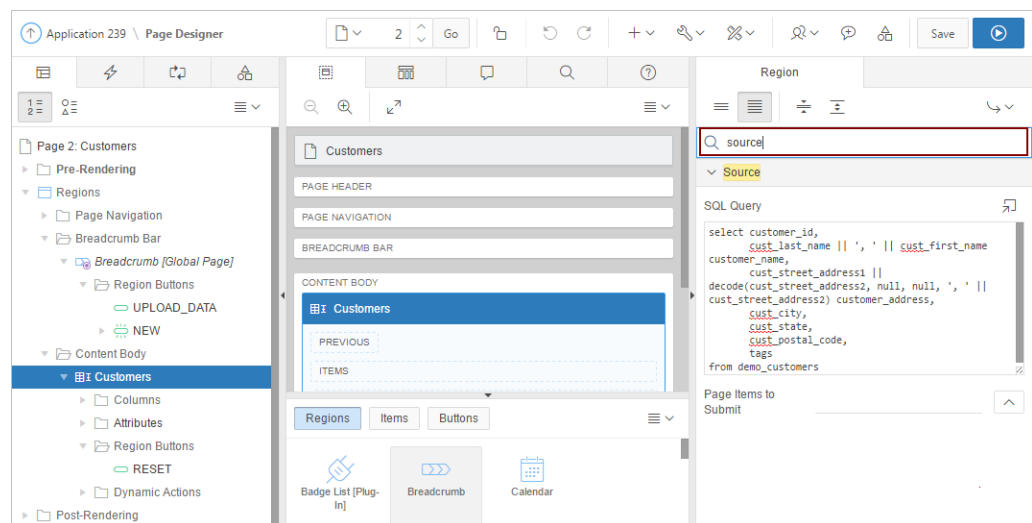
The right pane in Page Designer contains the Property Editor. Property Editor displays all attributes for the current component. Clicking in the Component View tab, the Messages tab or the Page search tab also changes the selection in property editor. When you select multiple components, the Property 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. Use the following icons to expand or collapse these groups:

- **Show Common** - Displays only those attributes for which the default value has changed or are commonly edited.
- **Show All** - Displays all attributes (default).
- **Collapse All** - Collapses functional group so only the group heading displays.
- **Expand All** - Expands functional groups so all attributes displays.
- **Go to Group** - Navigate to and expand, if necessary, the selected group.

To search for a group or an attribute, enter a keyword in the Filter Properties field. The groups and attributes containing the term appear.



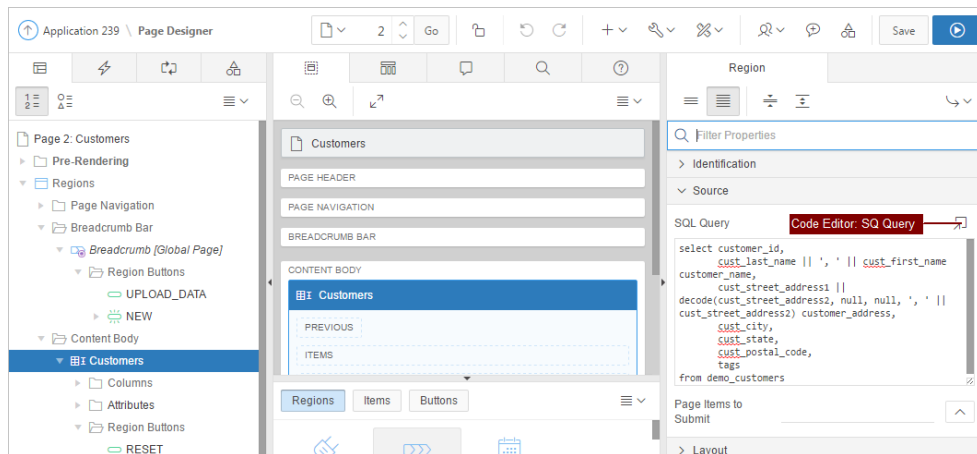
 **See Also:**

"Using Page Designer (page 7-21)" and "Managing Page Attributes in Page Designer (page 7-31)"

7.2.5 Code Editor

For attributes requiring large amounts of code (for example, a Region Source), you can:

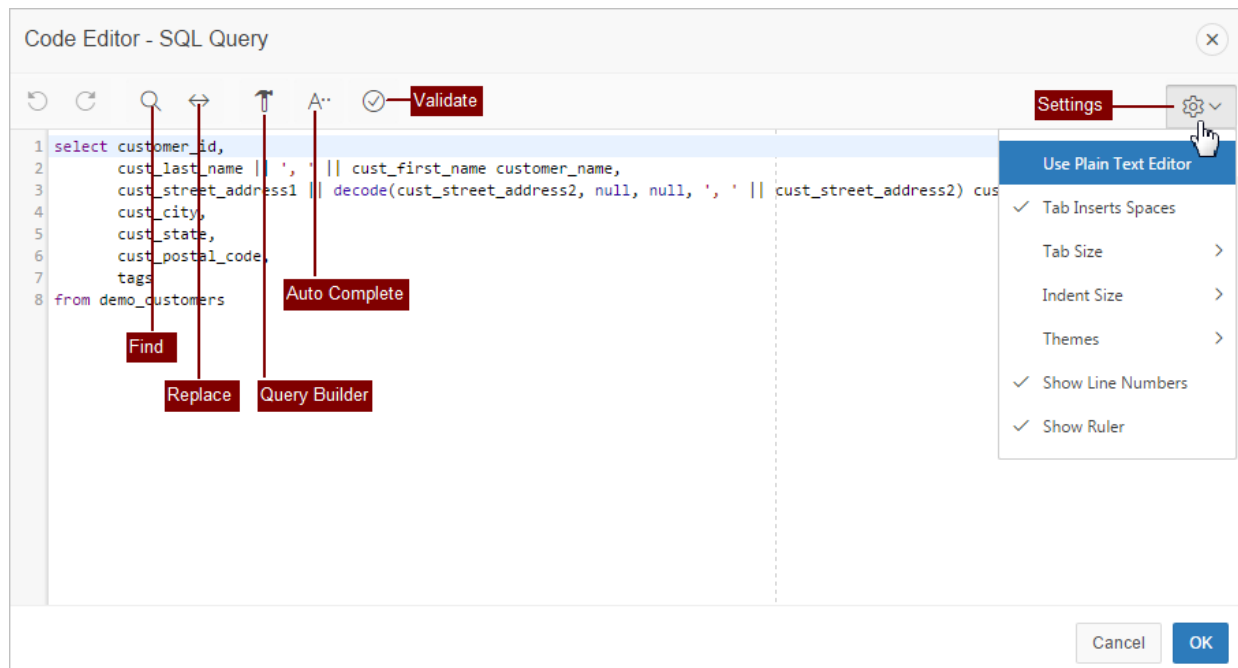
- Enter text directly in to the field.
- Click **Code Editor** to enter text a full featured code editor.



 **See Also:**

"Accessing Property Editor (page 7-24)"

A Code Editor displays as a modal dialog. You use the Code Editor to edit attributes requiring a large amount of code. The Code Editor 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
- Redo
- Find
- Replace
- Query Builder
- Auto Complete
- Validate
- Syntax highlighting
- Block indent and unindent
- Resize dialog

 **Tip:**

To switch to a plain text editor, select **Use Plain Text Editor** from the Settings menu.

7.3 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](#) (page 7-22)
- [Searching for Pages](#) (page 7-23)
- [Switching Between Three Pane and Two Pane Mode](#) (page 7-23)
- [Accessing Property Editor](#) (page 7-24)
- [Adding a Region, Item, or Button by Dragging and Dropping](#) (page 7-25)
- [About Adding Components Using Gallery Context Menus](#) (page 7-26)
- [Adding a Region, Item, or Button from the Rendering Tab](#) (page 7-27)
- [Adding a Region, Item, or Button from the Component View Tab](#) (page 7-28)
- [Copying a Region, Item, or Button](#) (page 7-29)
- [Creating Pre-Rendering Branches, Computations, and Processes](#) (page 7-30)
- [Creating Page Submission Branches, Validations, Computations, and Processes](#) (page 7-30)

 **See Also:**

["Understanding Page Designer UI Elements](#) (page 7-3)" and ["Viewing Help in Page Designer](#) (page 1-28)"

7.3.1 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.

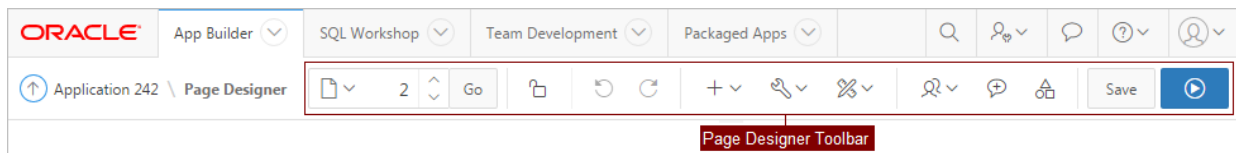
The default view, Page Designer appears.

You preview a page by running it. Running an application page displays the rendered version as end users would see it.

 **See Also:**

["Switching from Legacy Component View to Page Designer](#) (page 7-41)" and ["Running a Page from Page Designer](#) (page 7-37)"

7.3.2 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**.
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](#) (page 7-37)" and "[Creating Page Groups](#) (page 6-61)"

7.3.3 Switching Between Three Pane and Two Pane Mode

By default, Page Designer displays three panes. Use the options on the Setting menu on the Page Designer toolbar to switch between Two Pane Mode and Three Pane Mode. A typical use case for accessing Two Pane Mode, is when you want to view just the Rendering tree and the Property Editor.

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. On the Page Designer toolbar, click **Settings** and select one of the following:
 - a. **Two Pane Mode** — Displays just the central pane and the right pane.
 - b. **Three Pane Mode** — Displays the left pane, central pane, and the right pane.
 - c. **Reset Layout**— Returns the layout the default display. Select this option if you have moved tabs from their default positions or customized the tab size.

 **Tip:**

You can also drag and drop the tabs between panes. To return tabs to their default positions, select **Reset Layout** from the Settings menu on the Page Designer toolbar.

7.3.4 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.

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 \(Property Editor\)"](#) (page 7-18)" and *Oracle Application Express Accessibility Guide*

7.3.5 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:**

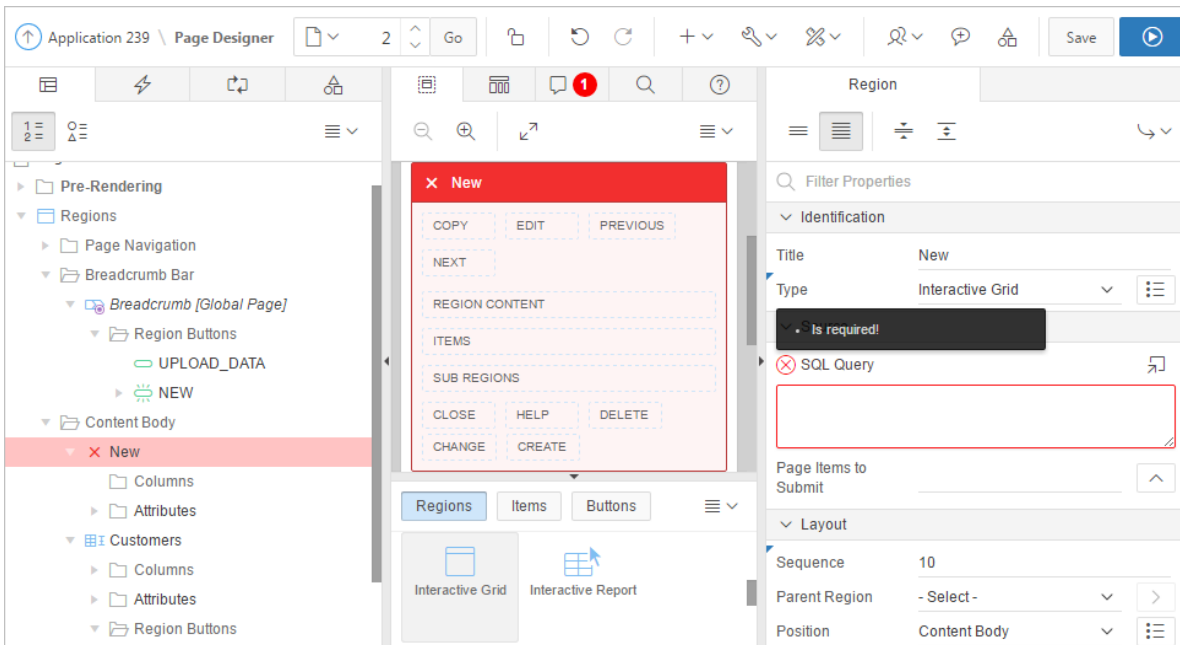
In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Select **Add To** and then the location where you want to add the component.

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 a SQL Query is required.



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.

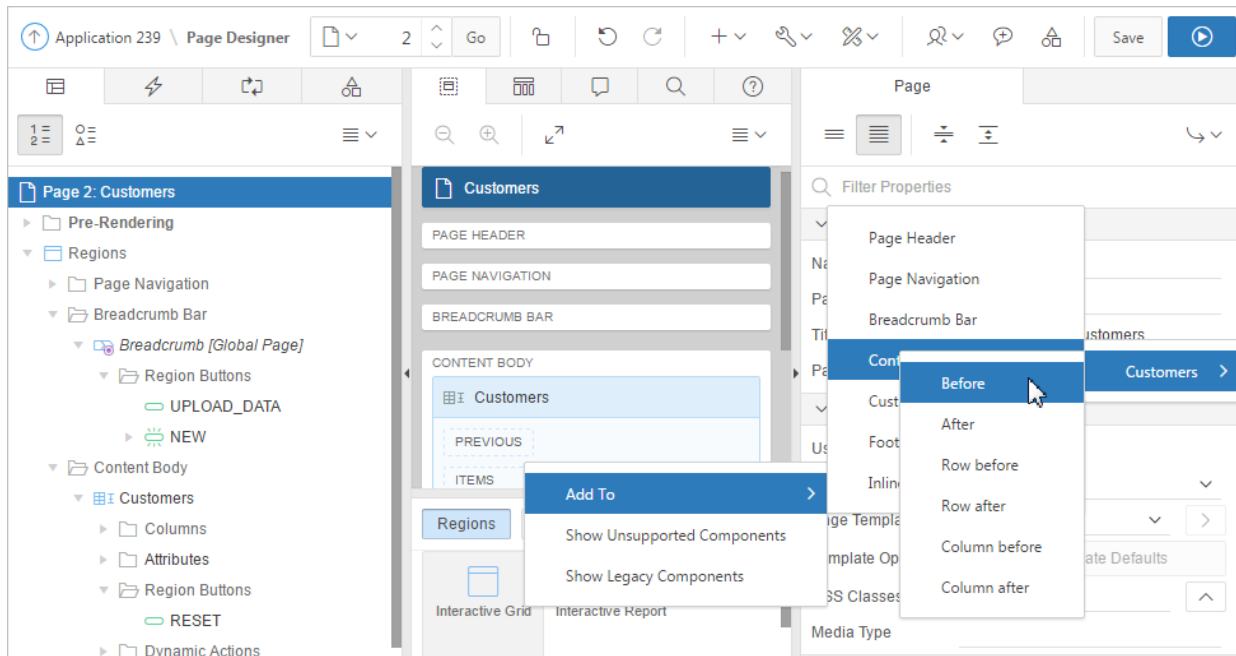
6. Click **Save** or **Save and Run Page**.

See Also:

- "[Messages Tab](#) (page 7-15)"
- "[Running a Page from Page Designer](#) (page 7-37)"
- "[About Adding Components Using Gallery Context Menus](#) (page 7-26)"

7.3.6 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 \(page 7-25\)"](#)

7.3.7 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 \(Property Editor\) \(page 7-18\)"](#)
- ["Managing Regions in Page Designer \(page 12-5\)"](#)
- ["Viewing Page-Level Items in Page Designer \(page 13-9\)"](#)
- ["Managing Buttons in Page Designer \(page 13-51\)"](#)

7.3.8 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 \(page 7-22\)"](#).

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:**

- ["Creating Buttons \(page 13-51\)"](#)
- ["Managing Regions in Page Designer \(page 12-5\),"](#)
- ["Viewing Page-Level Items in Page Designer \(page 13-9\),"](#)

7.3.9 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.

7.3.10 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 navigate to the Processing tab to maintain pre-rendering components.

 **See Also:**

["Managing Computations, Validations, and Processes \(page 15-1\)"](#)

7.3.11 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 \(page 15-1\)"](#)

7.4 Managing Page Attributes in Page Designer

Page attributes 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 in the Property Editor](#) (page 7-32)
- [Incorporating JavaScript Using Page Designer](#) (page 7-33)
- [Configuring Rejoin Sessions in Page Designer](#) (page 7-34)
- [Configuring Dialog Attributes in Page Designer](#) (page 7-35)
- [Configuring Page Attributes to Warn Users of Unsaved Changes](#) (page 7-36)



See Also:

["Right Pane of Page Designer \(Property Editor\) \(page 7-18\)"](#)

7.4.1 Editing Page Attributes in the Property Editor

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.

The Property Editor displays the Page attributes.

Property Editor organizes attributes into functional groups. Use the following icons to expand or collapse these groups:

- **Show Common** - Displays only those attributes for which the default value has changed or are commonly edited.
 - **Show All** - Displays all attributes (default).
 - **Collapse All** - Collapses functional group so only the group heading displays.
 - **Expand All** - Expands functional groups so all attributes displays.
 - **Go to Group** - Navigate to and expand, if necessary, the selected group.
3. To search for a group or an attribute, enter a keyword in the Filter Properties field.

The groups and attributes containing the term appear.
 4. 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.

5. To save your changes, click **Save** or **Save and Run Page**.



See Also:

["Right Pane of Page Designer \(Property Editor\) \(page 7-18\)"](#) and ["Viewing Field-Level Help \(page 1-27\)"](#)

7.4.2 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.

In the following example, adding the code would make the `test` function accessible from anywhere on the current page.

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. Click the **Collapse All** icon and then locate and expand **JavaScript**.
4. Under JavaScript, enter code into the following attributes:

 **Tip:**

To learn more about an attribute and view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.

- **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.
- **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:**

Note: Do not to 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.

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**.

7.4.3 Configuring Rejoin Sessions in Page Designer

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](#) (page 19-7)."

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. Click the **Collapse All** icon and then locate and expand **Security**.
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.

The option **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](#) (page 6-30)."

5. Click **Save**.

 **See Also:**

- "[Browser Security](#) (page 6-30)"
- "[About Rejoin Sessions](#) (page 19-7)"
- "Configuring Rejoin Sessions" in *Oracle Application Express Administration Guide*

7.4.4 Configuring Dialog Attributes in Page Designer

 **Tip:**

Creating a dialog page is a multiple step process. To learn more, see "[Creating Dialog Pages](#) (page 6-49)."

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. Click the **Collapse All** icon.
4. Locate and expand **Appearance** and verify that the Page Mode is set to either **Modal Dialog** or **Non-Modal Dialog**.
5. 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
6. To save your changes, click **Save** or **Save and Run Page**.



See Also:

"[Creating Dialog Pages](#) (page 6-49)" and [Right Pane of Page Designer \(Property Editor\)](#) (page 7-18)

7.4.5 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. Click the **Collapse All** icon and then locate and expand **Navigation**.

 **Tip:**

To search for an attribute, enter a keyword in the Filter Properties field. The attribute appears.

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 \(page 13-54\)"](#)
- ["Configuring Item Attributes to Warn Users of Unsaved Changes \(page 13-19\)"](#)
- *Oracle Application Express API Reference*

7.5 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:**

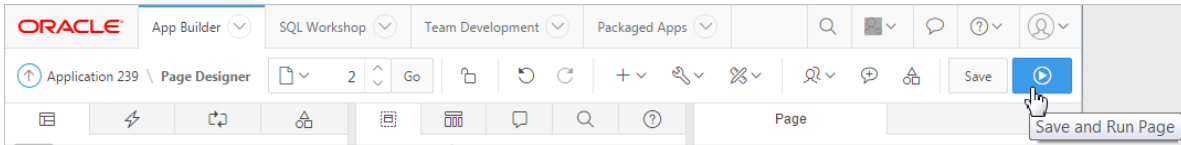
You cannot run Modal and Non-Modal dialog pages directly using the Save and Run Page button. To view a Modal and Non-Modal dialog page, run the normal page that is responsible for opening the dialog page and access it from there.

To run a page in Page Designer:

1. View the page in Page Designer. See "[Viewing a Page in Page Designer \(page 7-22\)](#)."

You preview a page by running it. Running an application page displays the rendered version as end users would see it.

- 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:

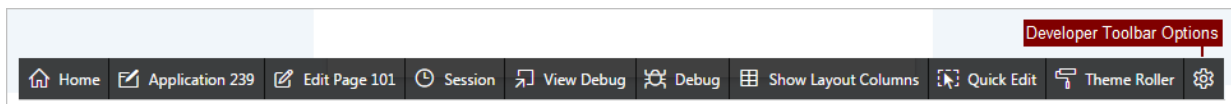
"[Running an Application or Page](#) (page 2-7)" and "[Runtime Developer Toolbar](#) (page 24-29)"

7.6 Runtime Developer Toolbar

When you run a desktop application and view a page, the Runtime Developer toolbar displays at the bottom of any editable running page. Use the Developer toolbar to quickly edit the current application or currently running page, or view session state.

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 of the following controls:

- Home** links to the Workspace home page. See "[Understanding the Workspace Home Page](#) (page 1-3)."
- Application** links to the Application home page. See "[Using the App Builder Home Page](#) (page 1-13)."

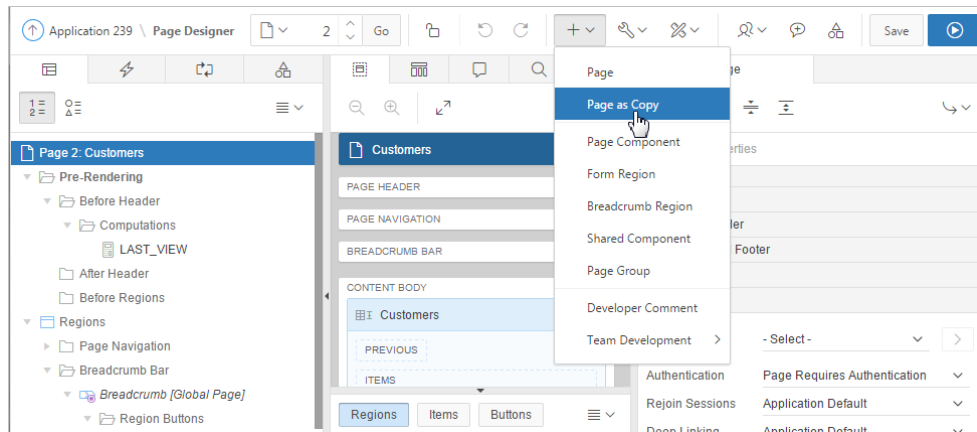
- **Edit Page** displays the currently running page ID.
 - **Session** links you to session state information for the current page. See "[Using the Session Page](#) (page 2-15)."
 - **View Debug** displays the Debug reports. See "[Using the Session Page](#) (page 2-15)."
 - **Debug** toggles the page between **Debug** and **No Debug** mode. See "[Utilizing Debug Mode](#) (page 22-4)."
 - **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.
 - **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](#) (page 11-40)."
- 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](#) (page 11-3)."
- **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)

7.7 Copying a Database Application Page in Page Designer

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.

7.8 Deleting Pages in Page Designer

Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries.

- [Deleting a Page in Page Designer](#) (page 7-40)
- [Deleting Multiple Pages](#) (page 7-41)
- [Deleting a Range of Pages](#) (page 7-41)

See Also:

"[Locking and Unlocking a Page](#) (page 6-64)"

7.8.1 Deleting a Page in Page Designer

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](#) (page 7-3)"

7.8.2 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**.

7.8.3 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**.

7.9 Switching from Legacy Component View to Page Designer

Component View is a legacy user interface for editing application pages. Oracle recommends developers use Page Designer instead of Legacy Component View. Certain components such as interactive grid and Oracle JET charts can not be maintained using the Legacy Component View. Unlike Page Designer, Legacy Component View 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, you must run it.

 **Tip:**

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. However, when you click a component and the Property Editor highlights the corresponding attribute in the right pane. See "[Component View Tab](#) (page 7-14)".

 **Important:**

Component View is disabled by default. To enable Component View, edit your user preferences. See "[Editing User Preferences](#) (page 1-23)".

To switch from Legacy Component View to Page Designer:

1. View the page in Component View.
2. To switch to Page Designer, click the **Page Designer** icon at the top of the page. Page Designer appears.
3. Select an application.

 **See Also:**

"[Viewing a Page in Legacy Component View](#) (page 24-10)" and "[Switching from Page Designer to Legacy Component View](#) (page 24-3)"

8

Developing Reports

In Oracle Application Express, a report is the formatted result of a SQL query. You can generate the SQL query by running a wizard or defining the SQL query manually.

Topics:

- [Understanding Report Types](#) (page 8-1)
- [Creating a Report](#) (page 8-5)
- [Managing Interactive Grids](#) (page 8-13)
- [Managing Interactive Reports](#) (page 8-38)
- [Managing Classic Reports](#) (page 8-101)
- [Printing Report Regions](#) (page 8-123)
- [Understanding BLOB Support in Forms and Reports](#) (page 8-130)



See Also:

"[Developing Forms](#) (page 9-1)" and "[Managing Database Application Components](#) (page 10-1)"

8.1 Understanding Report Types

When creating a database application, developers can create four basic report types: interactive grid, an interactive report, report on a form on a 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.

- [Interactive Grid](#) (page 8-1)
- [Interactive Reports](#) (page 8-2)
- [Classic Report](#) (page 8-4)
- [Report and Form](#) (page 8-4)

8.1.1 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, and sort individual columns with 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.

The screenshot shows an interactive grid with columns: Lead, Name, Description, Created, Completed, and Updated. An 'Actions' menu is open over the grid, showing options: Columns, Filter, Data (selected), Format, Report Settings, Download, and Help. The 'Data' sub-menu is also open, showing options: Sort, Aggregate, Flashback, and Refresh. The grid contains 5 rows of data.

Lead	Name	Description	Created	Completed	Updated
13	Configure Web Development Tool ...	Determine the hardwar...	26-AUG-16	30-JUL-16	26-AUG-16
1	Train Developers on Web develop...	Ensure all developers v...	14-AUG-16		26-AUG-16
10	Migrate Legacy Applications	Move the data and red...			26-AUG-16
13	Develop Partner Portal POC	Develop a proof of con...			26-AUG-16
1	Develop Production Partner Portal	Develop the productio...			26-AUG-16

The following is an example of an editable interactive grid. 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 screenshot shows an editable interactive grid with columns: Lead, Name, Description, Status Cd, Completed D..., Created, and Updated. A 'Row Actions' menu is open over the first row, showing options: Single Row View, Add Row, Duplicate Row, Delete Row, Refresh Row, and Revert Changes. The grid contains 5 rows of data.

Lead	Name	Description	Status Cd	Completed D...	Created	Updated
13	Configure Web Developme...	Determine the hardwar...	COMPLETED	30-JUL-16	26-AUG-16	26-AUG-16
	opers on Web d...	Ensure all developers ...	COMPLETED	14-AUG-16	26-AUG-16	26-AUG-16
	gacy Applications	Move the data and re...	IN-PROGRESS	-	26-AUG-16	26-AUG-16
	artner Portal POC	Develop a proof of co...	IN-PROGRESS	-	26-AUG-16	26-AUG-16
	roduction Partne...	Develop the productio...	ASSIGNED	-	26-AUG-16	26-AUG-16

See Also:

- *Oracle Application Express End User's Guide*
- ["Managing Interactive Grids \(page 8-13\)"](#)
- ["Making an Existing Interactive Grid Editable \(page 8-21\)"](#)

8.1.2 Interactive Reports

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.

Id	Name	Description	Project Start	Completed	Updated
1	Configure Web Development Tool Environment	Determine the hardware and software required to develop with Web Development tool.	26-AUG-16 01.55.37.479035 PM	30-JUL-16	26-AUG-16 01.55.37.479112 PM
2	Train Developers on Web development tool	Ensure all developers who are developing with the new tool receive appropriate training.	26-AUG-16 01.55.37.529162 PM	14-AUG-16	26-AUG-16 01.55.37.529272 PM
3	Migrate Legacy Applications	Move the data and redevelop applications currently running on legacy servers.	26-AUG-16 01.55.37.551825 PM	-	26-AUG-16 01.55.37.551897 PM
4	Develop Partner Portal POC	Develop a proof of concept that partners can use to work more collaboratively with us.	26-AUG-16 01.55.37.583695 PM	-	26-AUG-16 01.55.37.583785 PM
5	Develop Production Partner Portal	Develop the production application that partners can use to work more collaboratively with us.	26-AUG-16 01.55.37.598168 PM	-	26-AUG-16 01.55.37.598257 PM

1 - 5

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:

- ["Editing Interactive Reports in Page Designer \(page 8-38\) "](#)
- ["Editing Interactive Reports in Legacy Component View \(page 8-64\)"](#)
- ["Using Interactive Reports" in Oracle Application Express End User's Guide](#)

8.1.3 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.

Lead ↑	Name	Description	Status	Created	Completed	Updated
1	Develop Production Partner Portal	Develop the production app that partners can use to work more collaboratively with us.	ASSIGNED	26-AUG-16		26-AUG-16
1	Train Developers on Web development tool	Ensure all developers who will be developing with the new tool get the appropriate training.	COMPLETED	26-AUG-16	14-AUG-16	26-AUG-16
10	Migrate Legacy Applications	Move the data and redevelop the applications currently running on top of legacy servers	IN-PROGRESS	26-AUG-16		26-AUG-16
13	Develop Partner Portal POC	Develop a proof of concept that partners can use to work more collaboratively with us.	IN-PROGRESS	26-AUG-16		26-AUG-16
13	Configure Web Development Tool Environment	Determine the hardware and software required to develop with Web development tool.	COMPLETED	26-AUG-16	30-JUL-16	26-AUG-16

[Download](#)

1 - 5

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](#) (page 11-72)."



See Also:

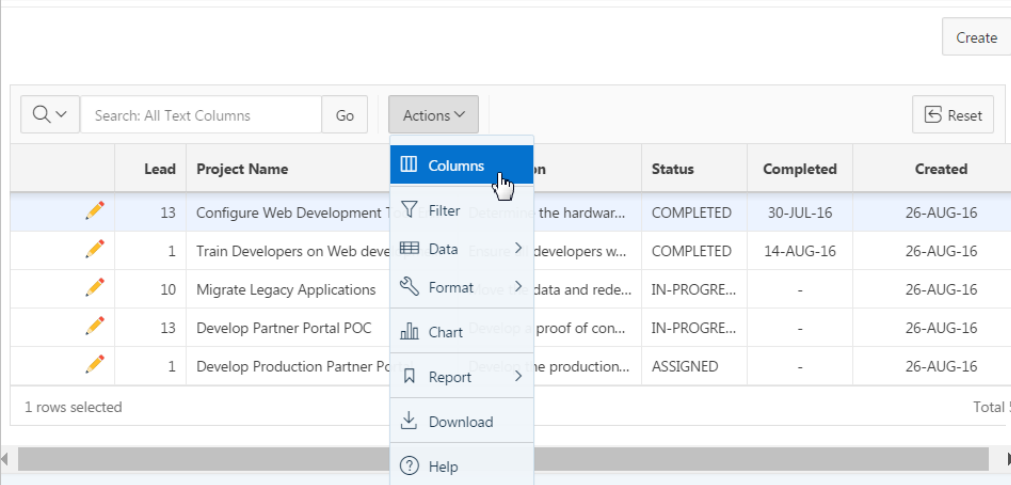
"[Editing Classic Reports in Page Designer](#) (page 8-102)" and "[Editing Classic Reports in Legacy Component View](#) (page 8-112)"

8.1.4 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. The Create Application Wizard enables the developer to select the report type (that is, interactive grid, interactive report, or classic) and the table. In addition to selecting the report type, the Create Page Wizard also enables the developer to 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.



Lead	Project Name	Status	Completed	Created
13	Configure Web Development	COMPLETED	30-JUL-16	26-AUG-16
1	Train Developers on Web deve	COMPLETED	14-AUG-16	26-AUG-16
10	Migrate Legacy Applications	IN-PROGRE...	-	26-AUG-16
13	Develop Partner Portal POC	IN-PROGRE...	-	26-AUG-16
1	Develop Production Partner PC	ASSIGNED	-	26-AUG-16

See Also:

- ["Interactive Grid \(page 8-1\)"](#)
- ["Interactive Reports \(page 8-2\)"](#)
- ["Classic Report \(page 8-4\)"](#)

8.2 Creating a Report

Developers can create reports by running a wizard, or by dragging and dropping the region type in Page Designer.

- [Creating Reports Using a Wizard \(page 8-5\)](#)
- [Creating a Report by Dragging and Dropping from the Gallery \(page 8-11\)](#)

8.2.1 Creating Reports Using a Wizard

App Builder includes many built-in wizards that enable you to generate reports. Developers can create reports using the Create Application Wizard, Create Page Wizard, and Create Region Wizard. Note that the available report options differ depending the selected user interface (for example, Desktop and Mobile).

- [Report Options When Running the Create Application Wizard \(page 8-6\)](#)

- [Report Options When Running the Create Page Wizard](#) (page 8-7)
- [Creating a Report Using the Create Application Wizard](#) (page 8-9)
- [Creating a Report Using the Create Page Wizard](#) (page 8-11)

8.2.1.1 Report Options When Running the Create Application Wizard

When you run the Create Application Wizard, you choose a target user interface. The user interface you select determines the available page types and report options.

Available Reports for Desktop Applications

If you are creating a desktop application, the Create Application Wizard supports the following report types.

Table 8-1 Create Application Wizard - Available Reports for Desktop Applications

Report Type	Description
Report	Creates a page that contains the formatted result of a SQL query. Developers choose a table on which to build a report, or provide a custom SQL SELECT statement. Once developer select a page source (that is Table or SQL Query) they choose a Report Type (interactive report , interactive grid , or classic report).
Report and Form	Choose a table on which to build a two page report and form combination. You then choose the report type (interactive report , interactive grid , or classic report). On the first page, users select a row to update. On the second page, users can add a record, or update or delete an existing record.
Editable Interactive Grid	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 or keyboard. You choose a table on which to build the interactive grid.

Available Reports for Mobile Applications

If you are creating a mobile application, the Create Application Wizard supports the following report types.

Table 8-2 Create Application Wizard - Available Report Types for Mobile Applications

Page Type	Description
List View	Creates a responsive report for mobile applications. The report works by collapsing the table columns into a stacked presentation that looks like blocks of label and data pairs for each row. This switch occurs when there is not enough space available to display the report horizontally.

Table 8-2 (Cont.) Create Application Wizard - Available Report Types for Mobile Applications

Page Type	Description
Column Toggle Report	Creates a responsive report for mobile applications. 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 display at narrower widths (for example, columns with priority 6 will be hidden first and columns with priority 1 will be hidden last.) The report includes a Columns button which enables end users to select which columns they want to view.
Reflow Report	Creates a responsive report for mobile applications. When there is not enough space available to display the report horizontally, the report works by collapsing the table columns into a stacked presentation that looks like blocks of label and data pairs for each row.
List View	Displays the results of a SQL query in a list for mobile applications.
List View and Form	Optimized to display data and provide easy navigation on Smartphones. Builds a two page List view and form combination. On the first page, users select a row to update from the List View. On the second page, users can update the selected table or view.

 **See Also:**

"[Creating a Report Using the Create Application Wizard \(page 8-9\)](#)", "[Available Page Types When Running the Create Application Wizard \(page 6-2\)](#)", and "[Developing Reports \(page 8-1\)](#)"

8.2.1.2 Report Options When Running the Create Page Wizard

When you create an application, you choose a target user interface. The user interface you select determines the available report types.

Available Reports for Desktop Applications

If you are creating a desktop application, the Create Page Wizard supports the following reports.

Table 8-3 Create Page Wizard — Reports for Desktop Applications

Report Type	Description	To Learn More
Interactive Report	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. End users can customize the report layout and data displayed by selecting options on the Actions menu.	See " Managing Interactive Reports (page 8-38)"
Interactive Grid	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 SELECT statement. 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. 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.	See " Managing Interactive Grids (page 8-13)" or " Making an Existing Interactive Grid Editable (page 8-21)"
Classic Report	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.	See Managing Classic Reports (page 8-101)
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.
Report on Web Service	Creates a report on a Web Service result.	"See Implementing Web Services (page 17-8)"

Available Reports for Mobile Applications

If you are creating a mobile application, the Create Page Wizard supports the following reports.

Table 8-4 Create Page Wizard — Supported Reports for Mobile Applications

Page Type	Description	To Learn More
List View	List View is optimized 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 (page 8-1)"
Column Toggle Report	Creates a responsive report for mobile applications. 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 (page 8-1)"
Reflow Report	Creates a responsive report for mobile applications. When there is not enough space available to display the report horizontally, the report works by collapsing the table columns into a stacked presentation that looks like blocks of label and data pairs for each row.	See " Developing Reports (page 8-1)"
Report on Web Service	Creates a report on a Web Service result.	See " Implementing Web Services (page 17-8)"

8.2.1.3 Creating a Report Using the Create Application Wizard

By running the Create Application Wizard, you can create a new application that contains one or multiple 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. Select the application type and click **Next**.
4. For Name:
 - a. Schema - Each application obtains its privileges by parsing all SQL as a specific database schema. Identify the database schema owner.
 - b. Name - Enter the name used to identify the application to developers.
 - c. Application - Specify an integer value that identifies the application. The application ID must be unique within your Application Express instance. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - d. Theme - Select a theme. Themes are collections of templates used to define the layout and style of an entire application.

To view available themes, click **Select Theme**. Standard themes display. To view additional themes, make a selection from the Show list. When you select a theme the application automatically subscribes to it.

- e. Theme Style - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51.
- f. Click **Next**.

 **Tip:**

To create an instant application that includes just a login page and blank page, click the **Create Application** button.

Next, add pages to your application.

5. For Pages, click **Add Page**.
 - a. Select Page Type - Select the type of page you want to add. See "[Report Options When Running the Create Application Wizard](#) (page 8-6)."

Based on the Page Type, the wizard prompts you for different information, such as selecting a table name or report implementation. To learn more, see field-level Help. **Action** displays the currently selected page type. For each selection, the wizard prompts you for different types of information, such as selecting a table name or report implementation. To learn more, see field-level Help.
 - b. Click **Add Page**.

The page (or pages) appear at the top of the page. To delete a page, click the **Delete** icon (the X in the far right column).
 - c. Repeat the previous steps until all pages have been added.

 **Tip:**

To exit the wizard early and use App Builder Defaults, click the **Create Application** button.

- d. Click **Next**
6. When you are done adding pages, click **Next**.
7. Follow the on-screen instructions.

 **See Also:**

- "[Using Themes](#) (page 11-1)"
- "[Understanding Theme Subscriptions](#) (page 11-17)"
- "[Using Theme Styles](#) (page 11-3)"
- "[Creating a Database Application](#) (page 6-4)"

8.2.1.4 Creating a Report Using the Create Page Wizard

By running the Create Page Wizard, you can add a new page containing a report to an existing application.

To create a report using the Create Page 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.
 - b. Select a page type - Select **Report**.
 - c. Click **Next**.
5. Select a report type. See "[Report Options When Running the Create Page Wizard](#) (page 8-7)."

The report options that display depend upon the application user interface (for example, Desktop or jQuery Mobile Smartphone).

6. Click **Next**.
7. Follow the on-screen instructions.

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

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 and Component View by clicking the **Create** menu and selecting **Page**.

See Also:

["Managing Pages in a Database Application](#) (page 6-44)" and "[Viewing Field-Level Help](#) (page 1-27)"

8.2.2 Creating a Report by Dragging and Dropping 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:**

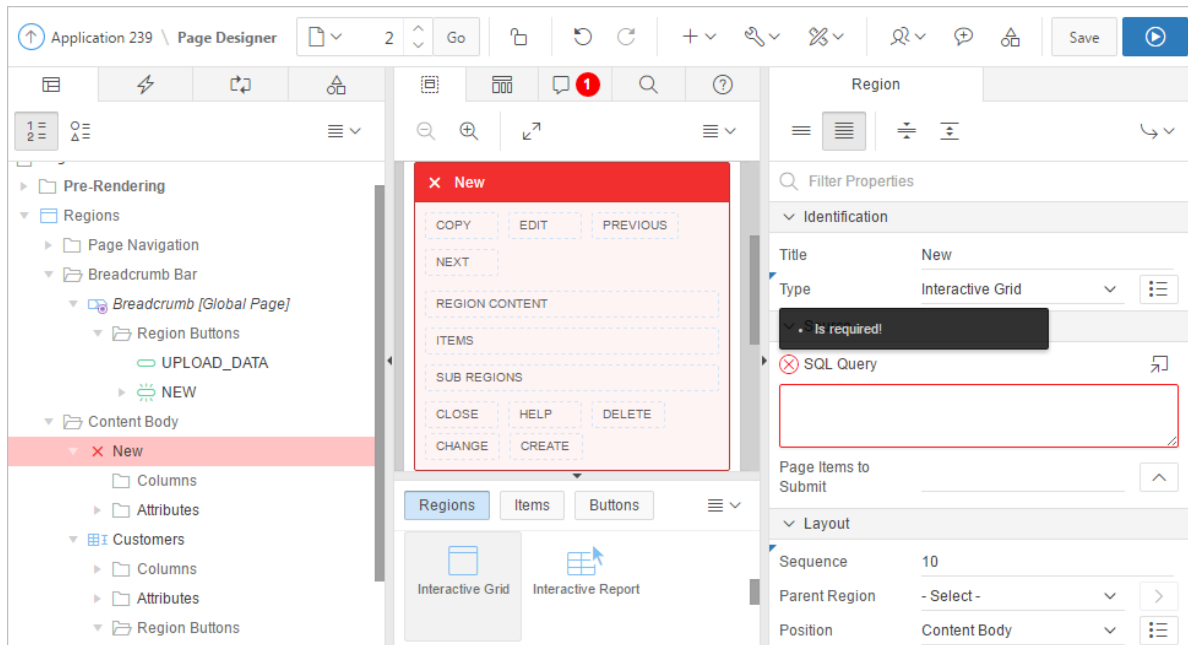
In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Select **Add To** and then the location where you want to add the component.

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 \(page 7-15\)"](#)

8.3 Managing Interactive Grids

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 a mouse or keyboard. With the Actions and Column Heading menus, users can hide, filter, freeze, highlight, sort, and create control breaks on individual columns. Advanced users can also define aggregations, which appear at the bottom of the column or column group.

Developers control how an interactive grid works by editing Region attributes, interactive grid Attributes, and Column attributes in Page Designer. For example, editing interactive grid Attributes determines whether the underlying data is read-only or editable by users.

 **Note:**

To learn more about features and capabilities of interactive grids, install the sample application, *Sample Interactive Grids*. See "[Installing a Packaged Application](#) (page 3-4)."

- [Managing Interactive Grid Region Attributes](#) (page 8-14)
- [Managing Interactive Grid Attributes](#) (page 8-20)
- [Managing Interactive Grid Column Attributes](#) (page 8-29)
- [About Customizing Interactive Grids in a Running Application](#) (page 8-33)
- [Saving Interactive Grids](#) (page 8-33)

 **See Also:**

"[Making an Existing Interactive Grid Editable](#) (page 8-21)"

8.3.1 Managing Interactive Grid Region Attributes

Each page in an Oracle Application Express application contains one or more regions. A region is an area on a page that serves as a container for content such as an interactive grid. Developers can 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](#) (page 8-14)
- [Creating Master Detail from an Existing Interactive Grid](#) (page 8-15)
- [Displaying Interactive Grids Conditionally](#) (page 8-19)

 **See Also:**

"[Managing Pages in Page Designer](#) (page 7-1)"

8.3.1.1 Editing Interactive Grid Region Attributes

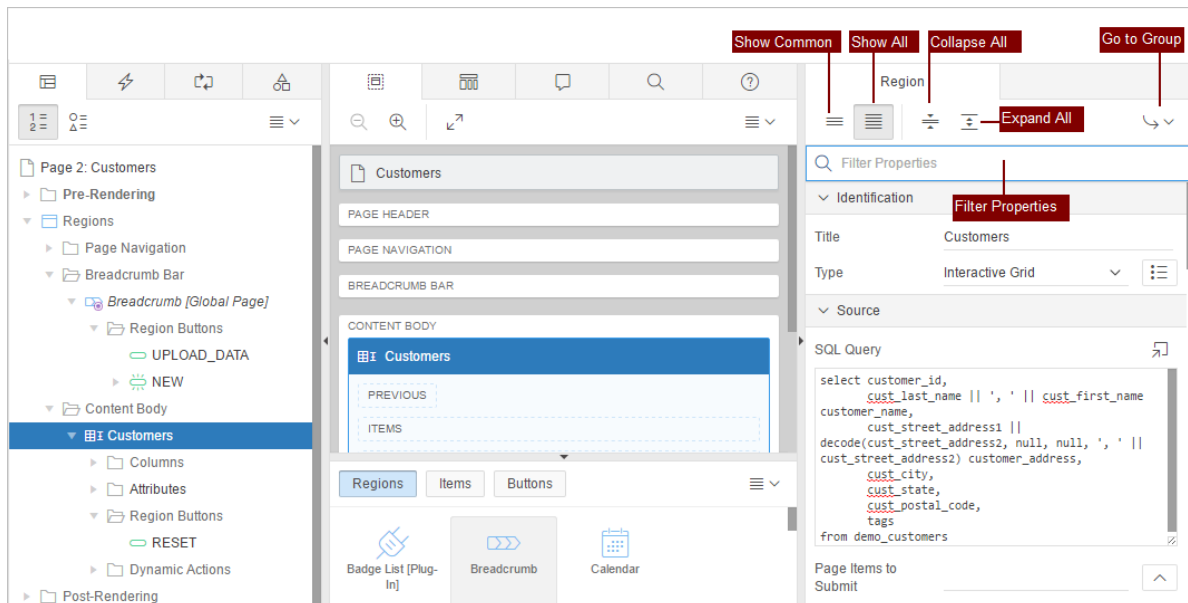
To edit region attributes in Page Designer:

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.

- In the Rendering tab, select the region.

The Property Editor displays the region attributes in the right pane.



Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

- To search for an attribute, enter a keyword search in the Filter Properties field. The groups and attributes containing the term appear.
- 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.

- To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

See Also:

"[Right Pane of Page Designer \(Property Editor\)](#) (page 7-18)"

8.3.1.2 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. 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, `EBA_DEMO_IG_DEPT` 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, `EBA_DEMO_IG_EMP`.

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 - Accept the defaults and click **Next**.
 - d. On Report Source:
 - Editing Enabled - Select **Yes**.
 - Table / View Name - Select **EBA_DEMO_IG_DEPT**.
 - Primary Key Column - Select **DEPTNO (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 `EBA_DEMO_IG_DEPT` table.

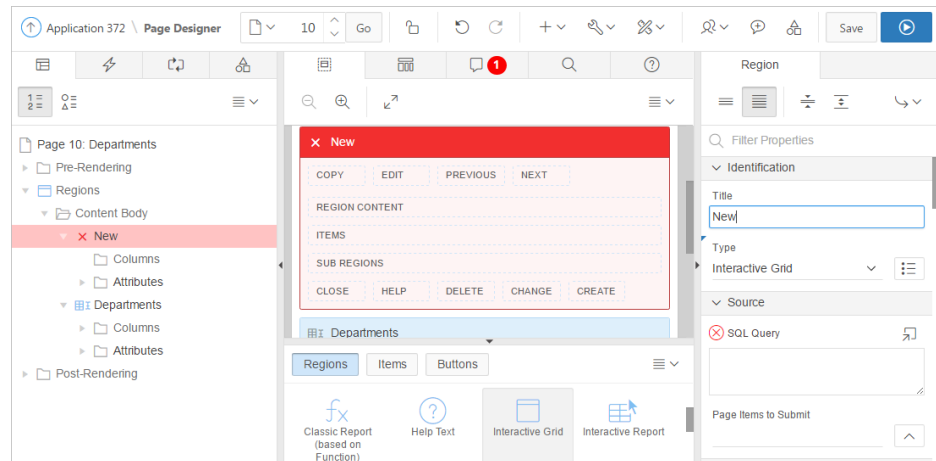
The screenshot shows a web interface for an interactive grid titled "Departments". At the top, there is a search bar with the text "Search: All Text Columns" and a "Go" button. To the right of the search bar are buttons for "Actions", "Edit", "Save", "Add Row", and "Reset". Below the search bar is a table with the following columns: a checkbox, a menu icon, "Dname", "Loc", and "Notes". The table contains four rows of data:

<input type="checkbox"/>	☰	Dname	Loc	Notes
<input checked="" type="checkbox"/>	☰	ACCOUNTING	NEW YORK	-
<input type="checkbox"/>	☰	RESEARCH	DALLAS	-
<input type="checkbox"/>	☰	SALES	CHICAGO	-
<input type="checkbox"/>	☰	OPERATIONS	BOSTON	-

At the bottom of the table, there is a status bar that says "1 rows selected" on the left and "Total 4" on the right.

5. Return to Page Designer. Click **Edit Page X** on the Runtime Developer toolbar.
6. Create the child interactive grid on the `EBA_DEMO_IG_EMP` table:
 - a. In the Gallery, locate the Regions tab.

- b. Under Regions, 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.
 - b. In the Property Editor, edit the region attributes:

 **Note:**

To search for an attribute, enter search terms in Filter Properties field at the top of the Property Editor.

- Identification, Title - Enter `Employees`.
- Source, SQL Query - Enter the query:

```
select EMPNO,
       ENAME,
       JOB,
       MGR,
       HIREDATE,
       SAL,
       COMM,
       ONLEAVE,
       DEPTNO
from EBA_DEMO_IG_EMP
```

- Layout, Start New Row - Select **Yes**.
 - Appearance, Template - Select **Standard**
 - Master Detail, Master Region - Select the region, **Departments**.
8. For the Child Interactive Grid, define `DEPTNO` as the Master Column:
 - a. In the Rendering tab, under Child Interactive Grid, expand the Columns node and select **DEPTNO**.
 - b. In Property Editor, edit the following:

- Master Detail, Master Column - Select **DEPTNO**.

9. Click **Save**. To run the page, click **Save and Run Page**.

Departments

		Dname	Loc	Notes
<input checked="" type="checkbox"/>	☰	ACCOUNTING	NEW YORK	-
<input type="checkbox"/>	☰	RESEARCH	DALLAS	-
<input type="checkbox"/>	☰	SALES	CHICAGO	-
<input type="checkbox"/>	☰	OPERATIONS	BOSTON	-

1 rows selected Total 4

Employees

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Onleave
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	N
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	N
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	N

Note that parent, Department, filters the results in Employees, the child.

Next, make the child, Employees, editable.

10. To make the child, Employees 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. Click **Save**. To run the page, click **Save and Run Page**.

Departments

Search: All Text Columns Go Actions Edit Save Add Row Reset

<input type="checkbox"/>		Dname	Loc	Notes
<input checked="" type="checkbox"/>		ACCOUNTING	NEW YORK	-
<input type="checkbox"/>		RESEARCH	DALLAS	-
<input type="checkbox"/>		SALES	CHICAGO	-
<input type="checkbox"/>		OPERATIONS	BOSTON	-

1 rows selected Total 4

Employees

Search: All Text Columns Go Actions Edit Save Add Row Reset

<input type="checkbox"/>		Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
<input type="checkbox"/>		7839	KING	PRESIDENT	-	17-NOV-81	5000	-
<input type="checkbox"/>		7782	CLARK	MANAGER	7839	09-JUN-81	2450	-
<input type="checkbox"/>		7934	MILLER	CLERK	7782	23-JAN-82	1300	-

The child interactive grid, Employees, is now editable

8.3.1.3 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.

3. In the Property Editor click the **Collapse All** icon and then locate and expand **Server-side Condition**.
4. 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.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.3.2 Managing Interactive Grid Attributes

Interactive grid Attributes control how an interactive grid works. For example, developers use these attributes to determine 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](#) (page 8-20)
- [Making an Existing Interactive Grid Editable](#) (page 8-21)
- [Controlling Interactive Grid Pagination](#) (page 8-22)
- [Customizing the Interactive Grid Toolbar](#) (page 8-23)
- [Enabling Users to Save Public Reports in Interactive Grids](#) (page 8-24)
- [Restricting Who Can Save Public Reports](#) (page 8-25)
- [Enabling Icon View in an Interactive Grid](#) (page 8-26)
- [Enabling Detail View in an Interactive Grid](#) (page 8-28)



See Also:

["Managing Pages in Page Designer](#) (page 7-1)"

8.3.2.1 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 in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. To search for an attribute, enter a keyword search in the Filter Properties field. The groups and attributes containing the term appear.
5. 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.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

["Right Pane of Page Designer \(Property Editor\) \(page 7-18\)"](#)

8.3.2.2 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 in the Property Editor.

4. In the Property Editor, click the **Collapse All** icon and then expand **Edit**.

When an interactive grid is editable, the system automatically creates a process to process the records. When Enabled is set to **Yes**, additional options appear.

Configure the following Edit options:

- a. **Enabled** - Select **Yes**.
- b. **Allowed Operations** - Select supported data manipulation operations. Options include: **Add Row**, **Update Row**, **Delete Row**.
- c. **Allowed Row Operations Column** - To control the allowed operations on a per row basis add a column to your SQL query and select that column here. The column value must be **U** to allow update, **D** to allow delete or **UD** to allow both.
- 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.

Options include:

- **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. **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 first user to save changes. To prevent this from happening, updates can check the existing column value(s) in the database with the value(s) retrieved when the record was originally queried.
 - f. **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 \(Property Editor\) \(page 7-18\)"](#)

8.3.2.3 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 in the Property Editor.

4. In the Property Editor, click the **Collapse All** icon and then expand **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.

Configure the following Pagination 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.
5. 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 Packaged Application](#) (page 3-4)."

8.3.2.4 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](#) (page 8-21)".

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 in the Property Editor.
4. In the Property Editor, click the **Collapse All** icon and then expand **Toolbar**.

 **Tip:**

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

5. For Show, select **Yes** or **No** to determine if a toolbar displays above the report.
6. 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.
7. **Add Button Label** only displays when the interactive grid is editable. Enter an alternative label for the Add button. The default label is **Add Row**.
8. 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 Packaged Application](#) (page 3-4)."

8.3.2.5 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](#) (page 8-25)."

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 in the Property Editor.
4. In the Property Editor, click the **Collapse All** icon and then expand **Enable Users To**.
5. For Save Public Report, select **Yes**.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.3.2.6 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:

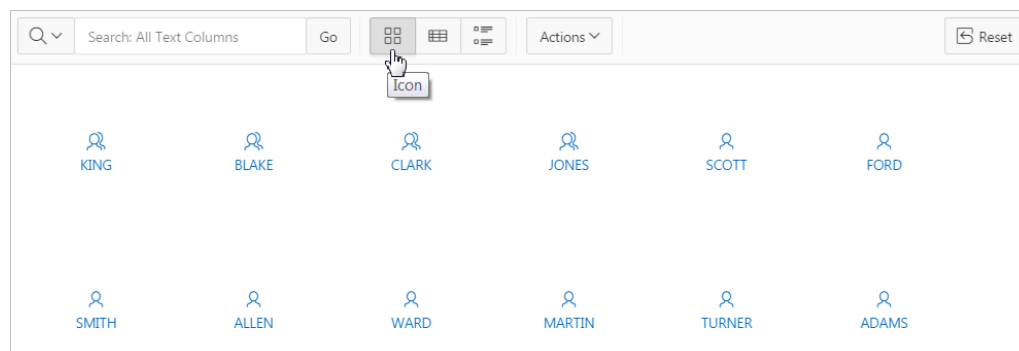
1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.Page Designer appears.
2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.
The Property Editor displays the Attributes in the Property Editor.
4. In the Property Editor, click the **Collapse All** icon and then expand **Enable Users To**.
5. For Save Public Report, select **Yes**.
6. For 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.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.3.2.7 Enabling Icon View in an Interactive Grid

Interactive grids support different views of the data. The grid view displays by default and users can also define a chart view. Developers can optionally add an Icon button to the select view button group as shown in the following illustration.



To use Icon view, you must identify the columns used to identify the icon, the label, and the target (that is, the link).

Tip:

To see a working example of this view, install the sample application, *Sample Interactive Grids* and review *Reporting, Icon and Detail Views*. See "[Installing a Packaged Application](#) (page 3-4)."

- [Adding Icon View to an Interactive Grid](#) (page 8-26)
- [Adding a Custom Icon View Link to an Interactive Grid](#) (page 8-27)

8.3.2.7.1 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:

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 in the Property Editor.
4. In the Property Editor, locate and expand **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 **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 CSS - Select the column that holds the icon CSS class.
 - 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. Label Column - Select the column that contains the label to use for the icon item.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.3.2.7.2 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 in the Property Editor.
4. In the Property Editor, locate and expand **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 - 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**.
 - 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 CSS - Select the column that holds the icon CSS class.
 - e. Link Target - Click **Link Target** to display the Link Builder - Link Target dialog. Define the link target. The link target defined is followed when the user activates the icon link with mouse or keyboard.
 - f. Label Column - Select the column that contains the label to use for the icon item.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.3.2.8 Enabling Detail View in an Interactive Grid

Interactive grids support different views of the data. By default grid view is always included and a chart view can be defined by the user. You can optionally define a detail view, which 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	

 **Tip:**

To see a working example of this view, install the sample application, *Sample Interactive Grids* and review *Reporting, Icon and Detail Views*. See "[Installing a Packaged Application](#) (page 3-4)."

- [Adding a Detail View to an Interactive Grid](#) (page 8-29)

8.3.2.8.1 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 in the Property Editor.
4. In the Property Editor, locate and expand **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**.

8.3.3 Managing Interactive Grid Column Attributes

Column attributes enable developers to control the display, features, and behavior of interactive grid columns. Developers can use these 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](#) (page 8-33)."

- [Editing Grid Column Attributes](#) (page 8-30)
- [Defining a Column as a List of Values for Filtering](#) (page 8-31)
- [Creating a Column Link in an Interactive Grid](#) (page 8-32)

 **See Also:**

"[Managing Pages in Page Designer](#) (page 7-1)"

8.3.3.1 Editing Grid Column Attributes

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.
The Property Editor displays the attributes for that column. Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
4. To search for an attribute, enter a keyword search in the Filter Properties field. The groups and attributes containing the term appear.
5. 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.

6. To edit the column heading:
 - a. Locate and expand **Heading**.
 - b. Edit the Heading, Alignment, and Alternative Label attributes.
7. To control how the column value displays and for editable grids how the column value is edited:
 - a. Locate and expand **Identification**.
 - b. Select the Type.

Different settings are available depending on the selected Type. In 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.

8. To control how users can manipulate a column:
 - a. Locate and expand **Enable Users To**.
 - b. For Sort/Control Break/Aggregate/Group By, select **Yes** or **No**. Selecting **Yes**, enables users to use this column in sorts, control breaks, aggregates, and group by.

 **Tip:**

Enabling these operations can cause performance issues, especially on large data sets or where appropriate column indexes are not available.

9. 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) (page 7-18)"

8.3.3.2 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.
3. Select the desired column.

4. In the Property Editor, locate and expand **Column Filter**. Specify the following:
 - a. Enabled - Select **Yes** or **No** to specify whether users can filter this column.
 - 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. See Page Designer Help for more details.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.3.3.3 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.

 **Tip:**

To control the amount of information that displays, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** buttons.

4. Expand **Identification**. From Type, select **Link**.
5. Expand **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**.

8.3.4 About Customizing Interactive Grids in a Running Application

When running an application, users (that is, both end users and developers) 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 \(page 8-21\)"](#)
- ["Saving Interactive Grids \(page 8-33\)"](#)
- *Oracle Application Express End User's Guide*

8.3.5 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 Packaged Application](#) (page 3-4)."

- [How the User Type Effects Available Save Options](#) (page 8-34)
- [About Configuration Dependencies When Saving Interactive Grids](#) (page 8-35)
- [Saving a Private Interactive Grid](#) (page 8-35)
- [Saving a Public Interactive Grid](#) (page 8-36)
- [Saving or Renaming a Primary Interactive Grid](#) (page 8-36)
- [Saving an Alternative Interactive Grid](#) (page 8-37)
- [Renaming a Saved Interactive Grid Report](#) (page 8-37)
- [Deleting a Saved Interactive Grid Report](#) (page 8-37)
- [Resetting a Saved Interactive Grid Report](#) (page 8-38)

8.3.5.1 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](#) (page 8-24)."

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.

8.3.5.2 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 \(page 19-45\)"](#)
- ["Customizing the Interactive Grid Toolbar \(page 8-23\)"](#)
- ["Enabling Users to Save Public Reports in Interactive Grids \(page 8-24\)"](#)

8.3.5.3 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.

8.3.5.4 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](#) (page 8-24)".

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.

8.3.5.5 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.

8.3.5.6 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.

8.3.5.7 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.

8.3.5.8 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.

8.3.5.9 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.

8.4 Managing Interactive Reports

An interactive 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. Interactive reports are only supported on Desktop pages. End users can customize the report layout and data displayed by selecting options on the Actions menu.

- [Editing Interactive Reports in Page Designer](#) (page 8-38)
- [Editing Interactive Reports in Legacy Component View](#) (page 8-64)
- [Linking to Interactive Reports](#) (page 8-87)
- [Customizing Interactive Reports in a Running Application](#) (page 8-92)

8.4.1 Editing Interactive Reports in Page Designer

By default, an interactive report includes a search bar, an Actions menu, a Column Heading menu, and Edit icons in the first column of each row. Users can use these controls to alter the layout of report data by selecting columns, applying filters, highlighting, and sorting. Developers can customize an interactive report in Page Designer and configure how these controls display.

- [Editing Interactive Report Region Attributes in Page Designer](#) (page 8-39)

- [Editing Interactive Report Column Attributes in Page Designer](#) (page 8-40)
- [About Filtering on Conditional Links in Interactive Report Columns](#) (page 8-41)
- [Editing Interactive Report Attributes in Page Designer](#) (page 8-42)
- [Controlling Interactive Report Pagination in Page Designer](#) (page 8-43)
- [Customizing the Interactive Report Search Bar](#) (page 8-43)
- [Customizing the Actions Menu in Page Designer](#) (page 8-47)
- [Configuring Actions Menu Download Options in Page Designer](#) (page 8-47)
- [Managing Link Columns in Page Designer](#) (page 8-52)
- [Creating a Column Link in an Interactive Report](#) (page 8-56)
- [Displaying Interactive Reports Conditionally](#) (page 8-57)
- [Defining a Column as a List of Values in an Interactive Report](#) (page 8-57)
- [Formatting Columns in an Interactive Report](#) (page 8-58)
- [Enabling Icon View in Page Designer](#) (page 8-59)
- [Enabling Detail View in Page Designer](#) (page 8-61)
- [Configuring Advanced Attributes for Interactive Reports](#) (page 8-62)



See Also:

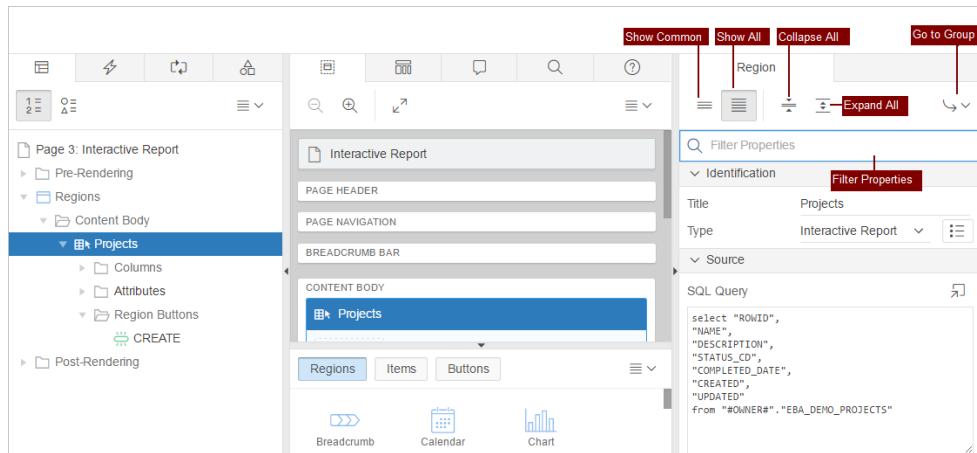
- ["Managing Pages in Page Designer](#) (page 7-1)"
- ["Linking to Interactive Reports](#) (page 8-87)"
- ["Customizing Interactive Reports in a Running Application](#) (page 8-92)"
- ["APEX_IR"](#) in *Oracle Application Express API Reference*

8.4.1.1 Editing Interactive Report Region Attributes in Page Designer

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.



Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

3. To search for a group or an attribute, enter a keyword in the Filter Properties field. The groups and attributes containing the term appear.
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 \(Property Editor\) \(page 7-18\)"](#)

8.4.1.2 Editing Interactive Report Column Attributes in Page Designer

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:

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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. To search for an attribute, enter a keyword search in the Filter Properties field.
The groups and attributes containing the term appear.
5. 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.

6. To edit the column heading:
 - a. Locate and expand **Heading**.
 - b. Edit the **Heading** and **Alignment** attributes.
7. To control how users can manipulate a column:
 - a. Locate and expand **Enable Users To**.
 - b. For Enable Users To, for each attribute select **Yes** or **No** to control how users can manipulate the report.
8. 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 \(page 8-41\)"](#) and ["Right Pane of Page Designer \(Property Editor\) \(page 7-18\)"](#)

8.4.1.3 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 in Page Designer \(page 8-40\)"](#)

8.4.1.4 Editing Interactive Report Attributes in Page Designer

Interactive report Attributes control how a report works. For example, you can use these attributes to configure pagination, create error messages, configure the Search bar, Action menu, and download options, control if and how users save the report, and configure supported views.

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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

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 \(Property Editor\) \(page 7-18\)"](#)

8.4.1.5 Controlling Interactive Report Pagination in Page Designer

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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Configure Pagination attributes:
 - a. Locate and expand **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**.

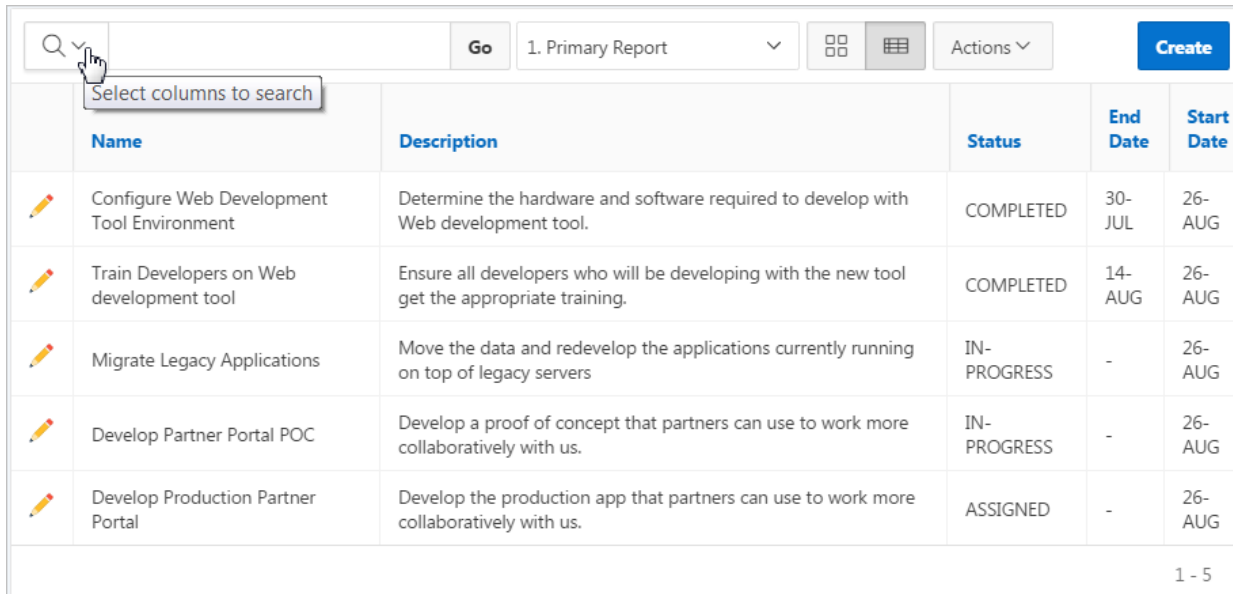
8.4.1.6 Customizing the Interactive Report Search Bar

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](#) (page 8-44)
- [Customizing the Search Bar in Page Designer](#) (page 8-44)
- [Removing the Interactive Report Search Bar in Page Designer](#) (page 8-46)

8.4.1.6.1 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.



	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 in Page Designer \(page 8-44\)"](#) and ["Removing the Interactive Report Search Bar in Page Designer \(page 8-46\)"](#)

8.4.1.6.2 Customizing the Search Bar in Page Designer

To customize the search bar:

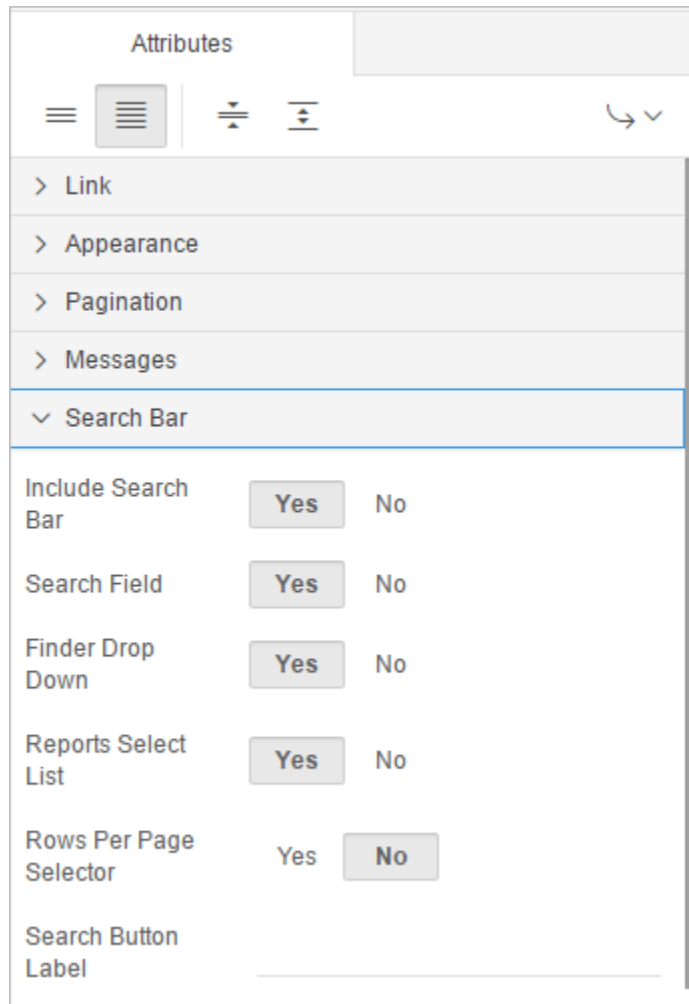
1. View the page in Page Designer:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.
 - c. Select a page.
 Page Designer appears.
2. In the Rendering tab, locate the region containing the report.

3. Under Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Locate and expand **Search Bar**.



The screenshot shows the 'Attributes' panel in the Oracle Property Editor. The 'Search Bar' attribute is expanded, showing the following options:

Include Search Bar	<input checked="" type="button" value="Yes"/>	<input type="button" value="No"/>
Search Field	<input checked="" type="button" value="Yes"/>	<input type="button" value="No"/>
Finder Drop Down	<input checked="" type="button" value="Yes"/>	<input type="button" value="No"/>
Reports Select List	<input checked="" type="button" value="Yes"/>	<input type="button" value="No"/>
Rows Per Page Selector	<input type="button" value="Yes"/>	<input checked="" type="button" value="No"/>
Search Button Label	<input type="text"/>	

5. Configure the options under Search Bar.

 **Tip:**

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

[Table 8-5](#) (page 8-46) describes the Search Bar attributes.

Table 8-5 Property Editor Search Bar Attributes

Attribute	Description
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	Displays a text field in the search bar used to enter search criteria. Select Yes or No .
Finder Drop Down	Embedded within the Search field, displays a drop down list of displayed columns. If a column is selected then the search is only performed against the values in the selected column. Select Yes or No . Within the interactive report, this option is identified as Select columns to search.
Reports Select List	Displays a list of all available reports for the user, including primary, alternate, public and private reports. This list is only displayed if a report, other than the primary report, is accessible to the user. Select Yes or No . See Also: " Saving Interactive Reports (page 8-96)".
Rows Per Page Selector	Displays a select list used to select the number of rows to display per page. Select Yes or No . See Also: " Saving Interactive Reports (page 8-96)".
Search Button Label	Specify the text to use as search button label.
Maximum Rows Per Page	Enter the maximum number to display in the Rows Per Page Selector.

- To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.6.3 Removing the Interactive Report Search Bar in Page Designer

To remove the search bar from in interactive report:

- View the page in Page Designer:
 - On the Workspace home page, click the **App Builder** icon.
 - Select an application.
 - Select a page.

Page Designer appears.

- In the Rendering tab, locate the region containing the report.
- Under Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

- Locate and expand **Search Bar**.
- For Include Search Bar, select **No**.
- To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.7 Customizing the Actions Menu in Page Designer

 **Tip:**

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](#) (page 16-11)".

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.

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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Locate and expand **Actions Menu**.
5. Configure the options under Actions Menu by selecting **Yes** or **No**.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.8 Configuring Actions Menu Download Options in Page Designer

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](#) (page 8-48)
- [Enabling CSV Download](#) (page 8-49)
- [Enabling Email Download](#) (page 8-50)

8.4.1.8.1 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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Collapse all attributes in the Property Editor. Click the **Collapse All** icon.
5. 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*.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

- "[Enabling CSV Download](#) (page 8-75)"
- "[Enabling Email Download](#) (page 8-75),"
- "[About Printing Reports to PDF](#) (page 8-124)"

8.4.1.8.2 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](#) (page 6-32)" and "[Automatic CSV Encoding](#) (page 6-35)."

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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

 **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**.

8.4.1.8.3 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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

 **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 Formats** - Select **Email**.

 **Tip:**

The Download option only appears on the Actions menu if a file format is selected.

7. 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 \(page 8-85\)"](#)

8.4.1.9 Managing Link Columns in Page Designer

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 \(page 8-52\)](#)
- [Linking to a Specific Page \(page 8-53\)](#)
- [Linking to a URL \(page 8-54\)](#)
- [Excluding a Link Column \(page 8-55\)](#)

8.4.1.9.1 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 report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.



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. Locate and expand **Link**.

- 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. Configure Single Row View attributes:
 - a. Locate and expand **Single Row View**.
 - b. Exclude Null Values - Specify whether null columns are hidden on the Single Record View
 - c. 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).`

8.4.1.9.2 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 report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

**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. Expand **Link**.
 - b. Link Column - Select **Link to Custom Target**.
 - c. Target -
 - Click the **Target** field.
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 - Accept the default or specify another image.
 - e. Link Attributes (Optional) - Specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.9.3 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 report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

 **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. Expand **Link**.
 - b. Link Column - Select **Link to Custom Target**.
 - c. Target -
 - Click the **Target** field.
The Link Builder - Target dialog appears.
 - Target Type - Select **URL**.
 - Click **OK**.
 - d. Link Icon - Accept the default or specify another image.
 - e. Link Attributes (Optional) - Specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.9.4 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 report attributes in the Property Editor.
Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

 **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. Expand **Link**.
 - 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**.

8.4.1.10 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

 **Tip:**

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

4. Expand Identification. From Type, select **Link**
5. Expand 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**.

8.4.1.11 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 the region containing the report.
3. In the Rendering tab, locate and select the region containing the report.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Configure Server-side Condition:
 - a. Locate and expand **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

5. 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.

8.4.1.12 Defining a Column as a List of Values in an Interactive Report

In interactive reports you can use 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 containing the report.
3. In the Rendering tab, expand the Columns node and select the desired column.
4. Configure Column Filter:
 - a. Locate and expand **Column Filter**.
 - b. For Type, select how to derive the filters provided in the report column drop down list. Options include:
 - **None** - Disable filter on list of values.
 - **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.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.13 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 containing the report.
3. In the Rendering tab, expand the Columns node and select the desired column.
4. Configure Column Filter:
 - a. Locate and expand **Column Formatting**.
 - b. In HTML Expression, specify HTML expressions to be shown for the 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

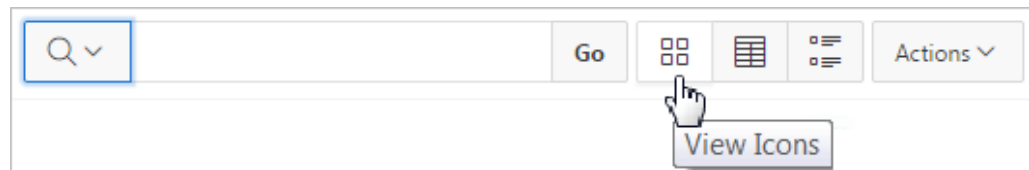
```

```

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.14 Enabling Icon View in Page Designer

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](#) (page 8-59)
- [Adding a Custom Icon View Link](#) (page 8-60)

8.4.1.14.1 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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Configure Icon View.
 - a. Locate and expand **Icon View**.
 - b. Configure 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**.
 - 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 \(page 8-60\)"](#)

8.4.1.14.2 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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Locate and expand **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:

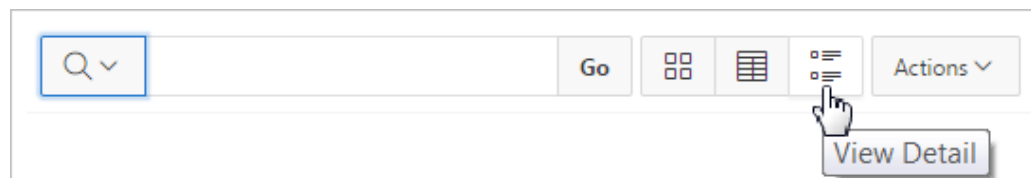

```
<a href="#EMP_LINK#"><br>#ENAME#</a>
```
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

See Also:

["Adding a Custom Icon View Link \(page 8-60\)"](#)

8.4.1.15 Enabling Detail View in Page Designer

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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Configure Detail View attributes:
 - a. Locate and expand **Detail View**.
 - b. Show - Select **Yes**.
 - c. Before Rows - Enter the HTML to be displayed before report rows. For example:


```
<table>
```
 - d. For Each Row - Enter the body for report rows in the Detail View. Use `#COLUMN_NAME#` substitution strings for column names and column labels. For example:


```
<tr><td class="u-textRight">#ENAME_LABEL#:</td><td>#ENAME#</td></tr>
<tr><td class="u-textRight">#JOB_LABEL#:</td><td>#JOB#</td></tr>
```
 - e. After Rows - Enter the HTML to be displayed after report rows. For example:


```
</table>
```
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.16 Configuring Advanced Attributes for Interactive Reports

By configuring advanced interactive report attributes, developers can define the region alias, specify an item whose value stores the saved report ID, and define a from address in an interactive report subscription or email download.

- [Configuring Advanced Attributes in Page Designer](#) (page 8-62)
- [About Emailing from an Interactive Report](#) (page 8-63)

8.4.1.16.1 Configuring Advanced Attributes in Page Designer

To configure Advanced 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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Locate and expand **Advanced**.

[Table 8-6](#) (page 8-63) describes the available Advanced Attributes.

Table 8-6 Advanced Attributes

Attribute	Description
Region Alias	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 <code>INTERACTIVE_REPORT_ID</code> by querying the <code>ALIAS</code> column in the <code>APEX_APPLICATION_PAGE_IR</code> view. Since the <code>INTERACTIVE_REPORT_ID</code> value can change when you export or import the application to different instance, referencing interactive report regions using an alias may be preferable.
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 <code>APEX_APPLICATION_PAGE_IR_RPT</code> view.
Email From Address	<p>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.</p> <p>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.</p> <p>Examples:</p> <pre>john.doe@abc.com &MY_APP_EMAIL_FROM. &APP_EMAIL.</pre>

- To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.1.16.2 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 \(page 6-16\)"](#)
- ["Configuring Advanced Attributes for Interactive Reports \(page 8-62\)"](#)
- ["Enabling Email Download \(page 8-50\)"](#)

8.4.2 Editing Interactive Reports in Legacy Component View

By default, an interactive report includes a search bar, an Actions menu, a Column Heading menu, and Edit icons in the first column of each row. Users can use these controls to alter the layout of report data by selecting columns, applying filters, highlighting, and sorting. Developers can customize an interactive report in Component View and configure how these controls display.

- [Editing Interactive Report Attributes in Component View \(page 8-65\)](#)
- [Editing Column Attributes in Component View \(page 8-67\)](#)
- [Creating Column Groups \(page 8-68\)](#)
- [Controlling Interactive Report Pagination in Component View \(page 8-68\)](#)
- [Managing the Search Bar and Actions Menu in Component View \(page 8-69\)](#)
- [Managing Link Columns in Component View \(page 8-76\)](#)
- [Creating a Column Link in an Interactive Report \(page 8-79\)](#)
- [Displaying Interactive Reports Conditionally \(page 8-80\)](#)
- [Defining a Column as a List of Values in an Interactive Report \(page 8-81\)](#)
- [Formatting Columns in an Interactive Report \(page 8-81\)](#)
- [Enabling Icon View in Component View \(page 8-82\)](#)
- [Enabling Detail View in Component View \(page 8-83\)](#)
- [Configuring Interactive Report Advanced Attributes \(page 8-84\)](#)
- [Editing Saved Default Interactive Reports in Component View \(page 8-86\)](#)
- [Viewing Saved Interactive Report Activity \(page 8-87\)](#)

 **See Also:**

"[Managing Pages in Legacy Component View](#) (page 24-1)," "[Linking to Interactive Reports](#) (page 8-87)," "[Customizing Interactive Reports in a Running Application](#) (page 8-92)," and "APEX_IR" in *Oracle Application Express API Reference*

8.4.2.1 Editing Interactive Report Attributes in Component View

Developers can customize an interactive report by editing report attributes.

- [Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)
- [Report Attribute Page Navigation Alternatives](#) (page 8-67)

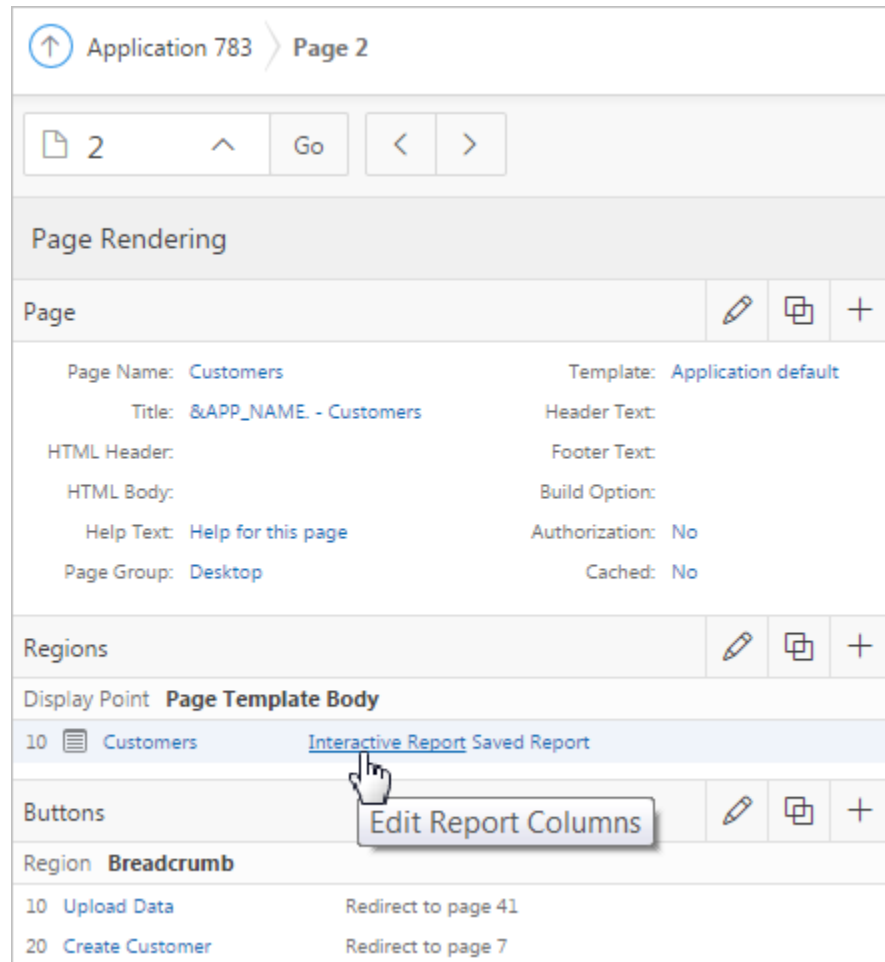
8.4.2.1.1 Accessing the Interactive Report Attributes Page in Component View

To access the Report Attributes page:

1. View the page in Component View. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."

Component View appears.

2. Under Regions, click the **Interactive Report** link next to the name of the report region you want to edit.



3. The Report Attributes page appears and is divided into the following sections:
 - **Column Attributes** control the report layout. See "[Editing Column Attributes in Component View](#) (page 8-67)."
 - **Column Groups** attributes enable you to group columns on the single row view. See "[Creating Column Groups](#) (page 8-68)."
 - **Pagination** attributes control report pagination. See "[Controlling Interactive Report Pagination in Component View](#) (page 8-68)."
 - **Search Bar** contain attributes that enable you to define the search bar within an interactive report. See "[Managing the Search Bar and Actions Menu in Component View](#) (page 8-69)."
 - **Download** enables you to add a download option to the Actions menu and export the report as CSV, HTML, Email, XLS, PDF, or RTF file. See "[Configuring Download Options on the Actions Menu](#) (page 8-73)."
 - **Link Column** attributes enable you to add a link to a column. See "[Managing Link Columns in Component View](#) (page 8-76)."
 - **Advanced Attributes** enable you to define the Interactive Report region alias, specify an item whose value stores the saved report id, and define items to set into session state. See "[Configuring Interactive Report Advanced Attributes](#) (page 8-84)."

- **Icon View** attributes enable you to add a View Icons icon to the Search bar. See "[Enabling Icon View in Component View](#) (page 8-82)."
- **Detail View** attributes enable you to add a View Details icon to the Search bar. See "[Enabling Detail View in Component View](#) (page 8-83)."
- **Description** lists a textual description of the current report.

 **Tip:**

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

 **See Also:**

"[Creating a Column Link in a Classic Report in Component View](#) (page 8-119)," "[Defining an Updatable Column in Component View](#) (page 8-120)," "[Defining a Column as a List of Values in a Classic Report in Component View](#) (page 8-121)," and "[Controlling When Classic Report Columns Display in Component View](#) (page 8-122)"

8.4.2.1.2 Report Attribute Page Navigation Alternatives

The Report Attribute page is divided into these sections. You can access these sections by 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**.

8.4.2.2 Editing Column Attributes in Component View

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, or hide a column.

To access Column Attributes:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears with the Column Attributes section at the top of the page.

Use the Column Attributes section to control report column appearance and functionality.

Heading displays the heading text to be included in the report. **Type** indicates the type of query column. The **Link** column indicates if a link column is currently defined.

[Table 8-7](#) (page 8-68) describes common report column edits.

Table 8-7 Common Report Column Edits for Interactive Reports

Description	Developer Action
Change column heading text.	Under Heading, enter different heading text.
Change column group assignment.	If Column Groups are defined, individual columns can be assigned under Column Group.
Change how columns display.	Select an option from Display Text As.

You can further refine the attributes of a specific column on the Column Attributes page.

2. To access the Column Attributes page, click the **Edit** icon next to the appropriate column.
The Column Definition appears.
3. Under Column Definition, Allow Users To, select the appropriate check boxes. Options include: Hide, Sort, Filter, Highlight, Control Break, Aggregate, Compute, Chart, and Group By. The options you select determine how users can manipulate the report.
4. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help.
5. Click **Apply Changes** to save your edits.

8.4.2.3 Creating Column Groups

Use column groups to group columns on a single row view.

To create a column group:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
2. Scroll down to Column Groups.
3. Click **Add Group**.
4. Under Column Group:
 - a. Sequence - Enter the display sequence for this column group. This attribute determines the order in which the column groups display in the detail view.
 - b. Group Name - Enter a column group name. This name displays in the region header in detailed view.
 - c. Description - Enter an optional description for the column group. This description never displays to the end user.
5. Click **Create**.

8.4.2.4 Controlling Interactive Report Pagination in Component View

You control how pagination displays by making selections from Pagination attributes on the Interactive Report Attributes page.

To access the Pagination section of the Report Attributes page:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Pagination.

You use the Pagination attributes to select a pagination style, determine where pagination occurs, specify the maximum number of rows to query, and messages to display when more than the maximum number of rows are returned or if no rows are returned.

[Table 8-8](#) (page 8-69) describes the available Pagination attributes.

Table 8-8 **Pagination Attributes**

Attribute	Description
Pagination Type	Select a pagination type. Pagination provides the user with information about the number of rows and the current position within the result set. Available pagination options include: <ul style="list-style-type: none"> • Row Ranges X to Y • Row Ranges X to Y of Z
Pagination Display Position	Defines where pagination occurs. Pagination can be display on the left side, right side, at the bottom, or above the report.
Show Null Values as	Enter the text you want to display for null columns. The default value is <code>(null)</code> .
Maximum Row Count	Defines the maximum number of rows to query, for example, rows 1 - 10 of 456. If you set this attribute to 200, the result would appear as follows: <code>rows 1 - 10 of more than 200 rows</code> This attribute impacts performance. Counting fewer rows can improve performance and counting thousands of rows can degrade performance.
When more than maximum row data found message	Defines the text message that displays when more than the maximum row count is returned.
When No Data Found Message	Defines the text message that displays when the query does not return any rows.

3. Click **Apply Changes** to save your edits.

8.4.2.5 Managing the Search Bar and Actions Menu in Component View

Developers can customize if and how the search bar and Actions menu display.

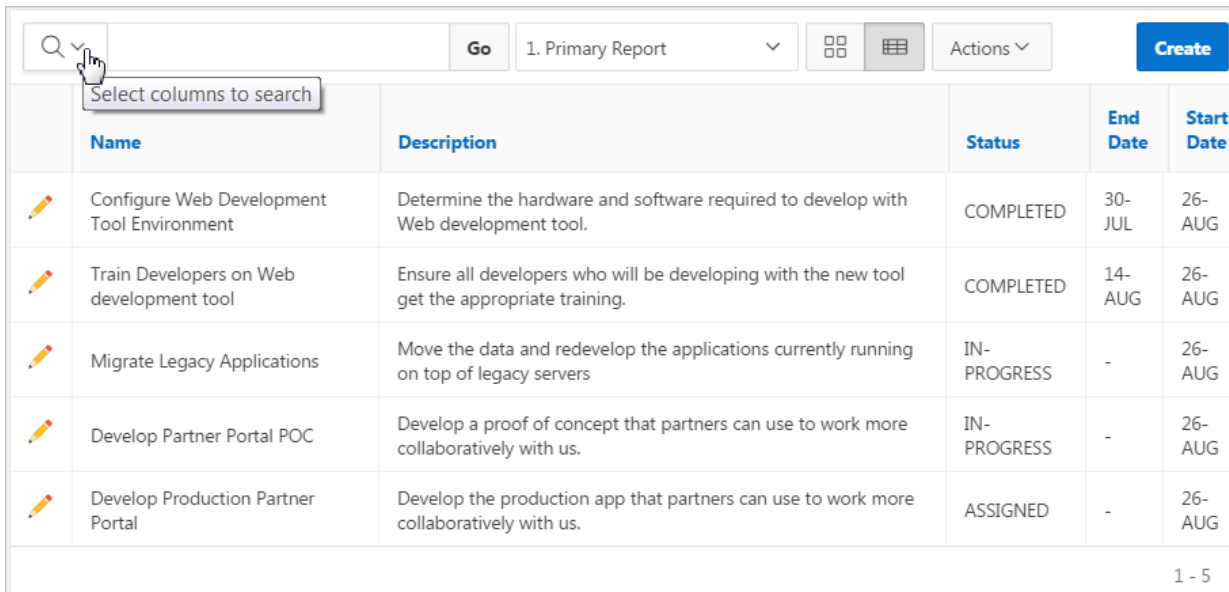
- [Interactive Report Search Bar and Actions Menu](#) (page 8-70)
- [Customizing the Search Bar and Actions Menu](#) (page 8-70)
- [Removing the Search Bar](#) (page 8-73)
- [Configuring Download Options on the Actions Menu](#) (page 8-73)






 **See Also:**

"[Enabling Icon View in Component View](#) (page 8-82)" and "[Enabling Detail View in Component View](#) (page 8-83)"

8.4.2.5.1 Interactive Report Search Bar and Actions Menu

A search bar displays at the top of every interactive report and includes a Select columns to search icon, Text area, Go button, and Actions menu.



	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. Developers can remove or customize the search bar by selecting and deselecting options in the Search Bar section of the Interactive Reports Attributes page.

8.4.2.5.2 Customizing the Search Bar and Actions Menu

To customize the search bar:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Search Bar.

Search Bar

Include Search Bar

Search Field
 Finder Drop Down
 Reports Select List
 Rows Per Page Selector
 Actions Menu

Include in Actions Menu (* for authenticated pages only):
 Select Columns
 Filter
 Rows Per Page
 Sort
 Control Break

Highlight
 Compute
 Aggregate
 Chart
 Group By

Pivot
 Flashback
 Save Report *
 Save Public Report *
 Reset

Help
 Download
 Subscription *

Save Public Report Authorization

Button Template

Finder Image

Actions Menu Image

Search Button Label

Maximum Rows Per Page

Use these attributes to customize how the search bar displays. [Table 8-9](#) (page 8-71) describes Search bar attributes:

Table 8-9 Search Bar Attributes

Attribute	Description
Include Search Bar	Select Yes or No . Selecting No removes the following default components: <ul style="list-style-type: none"> Search Field - Displays a text field in the search bar in which to enter search criteria. Finder Drop Down - Enables users to search on a selected column or against all columns. Reports Select List - Displays report select list if the report has alternative default, shared report or personal report. See "Saving Interactive Reports (page 8-96)." Rows Per Page Selector - Displays a select list to enable users to select the number of rows to display per page. Actions Menu - Enables the user to customize the report based on selected Actions.

Table 8-9 (Cont.) Search Bar Attributes

Attribute	Description
Include in Actions Menu	<p>Select and deselect functions to be included in the Actions menu. Available options include:</p> <ul style="list-style-type: none"> • Select Columns - Used to hide, unhide, and order the report columns. • Filter - Used to add filters to the report (for example, STATE = VA). • Rows Per Page - Used to display rows per page menu. • Sort - Used to sort columns. • Control Break - Used to build control breaks on report columns. • Highlight - Used to define conditions that highlights certain rows. • Compute - Used to add computed columns. • Aggregate - Used to include aggregate calculations on report columns. • Chart - Used to create a Flash chart based off the data in the report. • Group By - Used to create a Group By view based off the data in the report. • Pivot - Used to create a cross tab view based off the data in the report. • Flashback - Used to flashback the report data to some time in the past. • Save Report - Used to save the report settings for future use. Only available for authenticated pages. • Save Public Report - Used to save a public report for all users to view. You must select Save Report to enable this option. Only available for authenticated pages. • Reset - Used to reset the report to the default settings. • Help - Used to display instructions for using the Interactive report. • Download - Used to download the report in different formats. <p>Note: Disabling the Download functionality only prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all check boxes in the Download Formats section.</p> <ul style="list-style-type: none"> • Subscription - Used to schedule a subscription to send the report in email. Only available for authenticated pages. <p>See Also: "About the Actions Menu (page 8-94)".</p>
Save Report Public Authorization	<p>Users can save reports and classify them as being either public or private. Public reports can be viewed by all users. Private reports can only be viewed by the user who saved the report.</p> <p>In addition to enabling Save Public Report on the Actions Menu, you can add additional security by defining an Authorization Scheme to determine who has rights to enable Save Public Report.</p> <p>To enable support for public reports, select an authentication method from this list.</p>

Table 8-9 (Cont.) Search Bar Attributes

Attribute	Description
Search Button Label	Specify the text to use as search button label.
Maximum Rows Per Page	Enter the maximum number to display in the Rows Per Page Selector.

3. Click **Apply Changes**.

8.4.2.5.3 Removing the Search Bar

To remove the search bar:

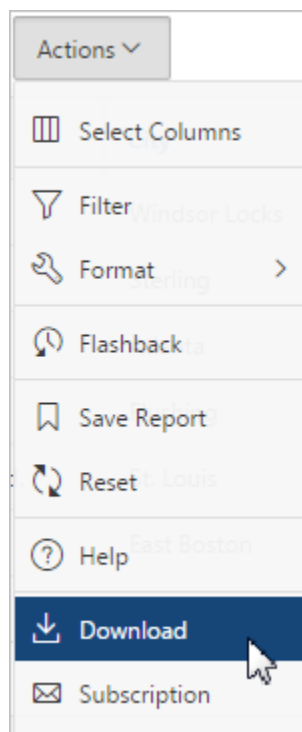
1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Search Bar.
3. From Include Search Bar, select **No**.
4. Click **Apply Changes**.

8.4.2.5.4 Configuring Download Options on the Actions Menu

The Download option to 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.



 **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](#) (page 8-74)
- [Enabling CSV Download](#) (page 8-75)
- [Enabling Email Download](#) (page 8-75)

8.4.2.5.4.1 Enabling Download Formats

To configure download formats:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Search Bar.
 - a. For Include Search Bar, select **Yes** and then select **Actions Menu**.
 - b. Under Include in Actions menu, select **Download**.

 **Tip:**

Disabling the Download functionality only prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all check boxes in the Download Formats section.

3. Scroll down to Download.
4. Select download formats. Options include:
 - **CSV**. See "[Enabling CSV Download](#) (page 8-75)."
 - **HTML**.
 - **Email**. See "[Enabling Email Download](#) (page 8-75)."
 - **XLS**.
 - **PDF**. See "[About Printing Reports to PDF](#) (page 8-124)."
 - **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. Click **Apply Changes**.

8.4.2.5.4.2 Enabling CSV Download

To enable the Enable CSV download:

1. Enable the CSV output option on the Edit Globalization Attributes page. See "[Accessing the Globalization Attributes Page](#) (page 6-32)" and "[Automatic CSV Encoding](#) (page 6-35)."

2. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

3. Scroll down to Search Bar.
 - a. For Include Search Bar, select **Yes** and then select **Actions Menu**.
 - b. Under Include in Actions menu, select **Download**.
4. Scroll down to Download.
5. From Download Formats, select **CSV**.

 **Tip:**

The Download option only appears on the Actions menu if a file format is selected.

6. In the CSV Separator, define the column separator. If no value is entered, a comma or semicolon is used depending on your current NLS settings.
7. In the 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 (" ").
8. In 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.
9. Click **Apply Changes**.

8.4.2.5.4.3 Enabling Email Download

You can configure the Download option to send 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 email download:

1. An Oracle Application Express administrator must configure email at the Instance level. See "Configuring Mail" in *Oracle Application Express Administration Guide*
2. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
3. Scroll down to Search Bar.
 - a. For Include Search Bar, select **Yes** and then select **Actions Menu**.
 - b. Under Include in Actions menu, select **Download**.
4. Scroll down to Download.
5. From Download Formats, select **Email**.

 **Tip:**

The Download option only appears on the Actions menu if a file format is selected.

6. Click **Apply Changes**.

 **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](#) (page 8-85)"

8.4.2.6 Managing Link Columns in Component View

A Link Column displays on the left side of an interactive report. This section describes how to 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](#) (page 8-77)
- [Linking to a Specific Page](#) (page 8-77)
- [Linking to a URL](#) (page 8-78)
- [Excluding a Link Column](#) (page 8-78)

8.4.2.6.1 Linking to a Single Row View

To link to a single row view:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Link Column.
3. For Link Column, select **Link to Single Row View**.
4. From Single Row View, select the appropriate options:
 - **Allow Exclude Null Values** - Excludes columns with null values.
 - **Allow Displayed Columns** - Displays only the columns shown in the report.
5. For Uniquely Identify Rows by, select **ROWID** or **Unique Column**.
If you select, **Unique Column**, specify the column in the Unique Column field.
6. For Link Icon, accept the icon specified or specify new image.
7. For Link Attributes (Optional), specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
8. Click **Apply Changes**.

 **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).`

8.4.2.6.2 Linking to a Specific Page

To link to a specific page:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
2. Scroll down to Link Column.
3. For Link Column, select **Link to Custom Target**.
4. For Link Icon, select an icon.
5. For Link Attributes (Optional), specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
6. For Target, select **Page in this Application**.
7. For Page, select the target page number. To reset the pagination for this page, select **Reset Pagination**.
8. For Request, specify the request to be used.
9. For Clear Cache, specify the pages (that is, the page numbers) on which to clear cache. You can specify multiple pages by listing the page numbers in a comma-delimited list. See "[Linking to Interactive Reports](#) (page 8-87)."
10. For Name and Value, specify session state for a specific item.
11. Click **Apply Changes**.

8.4.2.6.3 Linking to a URL

To link to a URL:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
2. Scroll down to Link Column.
3. For Link Column, select **Link to Custom Target**.
4. For Link Icon, select an icon.
5. For Link Attributes (Optional), specify additional column link attributes that to be included in the `` tag (for example, a link target, classes, or styles).
6. From Target, select **URL**.
7. In URL, enter the appropriate address.
8. For Checksum - Select one of the following:
 - Use default
 - Application Level - Link reusable by any user
 - User Level - Link reusable by current userTo learn more about an attribute, see field-level Help.
9. Click **Apply Changes**.

8.4.2.6.4 Excluding a Link Column

To exclude a column link:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
2. Scroll down to Link Column.
3. For Link Column, select **Exclude Link Column**.
4. Click **Apply Changes**.

8.4.2.7 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.

- [Creating a Column Link to Another Page](#) (page 8-79)
- [Create a Column Link to a URL](#) (page 8-80)

8.4.2.7.1 Creating a Column Link to Another Page

To create a column link to another page:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
2. Under Column Attributes, locate the column to contain the link.
3. Click the **Edit** icon next to the column name.
The Column Attributes page appears.
4. Scroll down to Column Link.
5. To create a column link to another page:
 - a. From Target, select **Page in this Application**.
 - b. (Optional) In Link Attributes, specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
 - c. In Link Text, enter the text to be displayed as a link, specify an image tag, or pick from the list of default images.
 - d. In Page, specify the target page number. To reset the pagination for this page, select **Reset Pagination**.
 - e. In Request, specify the request to be used.
 - f. In Clear Cache, specify the pages (that is, the page numbers) on which to clear cache. You can specify multiple pages by listing the page numbers in a comma-delimited list.
 - g. Use the Name and Value fields to specify session state for a specific item.
 - h. For Checksum - Select one of the following:
 - Use default
 - Application Level - Link reusable by any user
 - User Level - Link reusable by current user

To learn more, see field-level Help.

6. Click **Apply Changes**.

8.4.2.7.2 Create a Column Link to a URL

To create a column link to a URL:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
2. Access the Column Attributes page by clicking the **Edit** icon next to the appropriate column.
The Column Attributes page appears.
3. Scroll down to Column Link.
4. Under Column Link, specify the following:
 - a. From Target, select **URL**.
 - b. In Link Text, enter the text to be displayed as a link and select a substitution string.
 - c. (Optional) In Link Attributes, specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
 - d. In URL, enter the appropriate address.
5. Click **Apply Changes**.

8.4.2.8 Displaying Interactive Reports Conditionally

You can choose to have interactive reports display conditionally by editing region attributes on the Edit Region page.

To display an interactive report conditionally:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Region Definition. Under Regions, select the region name.
The Region Definition appears.
3. Scroll down to Conditions.
4. From the Condition Type list, make a selection.
5. Enter an expression in the fields provided.
6. Click **Apply Changes**.

 **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 an interactive report to function incorrectly.

8.4.2.9 Defining a Column as a List of Values in an Interactive Report

In interactive reports you can use a list of values to improve the speed of built-in filter tools.

To define a report column as a list of values:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Access the Column Attributes page by clicking the **Edit** icon next to the appropriate column.

The Column Attributes page appears.

3. Scroll down to List of Values.

4. From List of Values, Column Filter Type, specify how to create a list of values as a filter on the report column list. Available options include:

- **None** - Disable filter on list of values.
- **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.

5. Click **Apply Changes**.

8.4.2.10 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. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Access the Column Attributes page by clicking the **Edit** icon next to the appropriate column.

3. Scroll down to Column Formatting.

4. Specify HTML expressions to be shown for the column. Use substitution strings for column names to show column values in HTML. Consider the following examples:

Example 1:

```
<span style="background-color: yellow; color: red;">#EMPLOYEE_STATUS#</span>
```


Example 2:

```
<span class="#COLUMN_CLASS#">#EMPLOYEE_STATUS#</span>
```

In the previous example, the `#COLUMN_CLASS#` and `#EMPLOYEE_STATUS#` substitutions in this example are columns selected in the report SQL query.

Example 3:

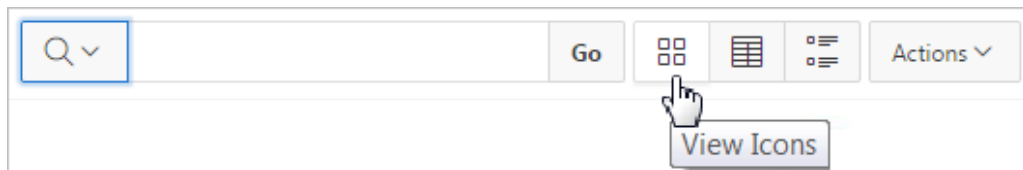
```

```

5. Click **Apply Changes**.

8.4.2.11 Enabling Icon View in Component 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 Icon View to an Interactive Report](#) (page 8-82)
- [Adding a Custom Icon View Link](#) (page 8-83)

8.4.2.11.1 Adding Icon View to an Interactive Report

To add Icon view to an interactive report:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."
The Interactive Report Attributes page appears.
2. Scroll down to Icon View.
3. For Icon View Enabled, select **Yes**.
4. Specify the following mandatory attributes:



Tip:

To view examples, see field-level Help.

- **Link Column** - Select the column that returns the link target of the icon.
- **Image Source Column** - Identify the column that returns the image source.

- Label Column - Select the column that returns the image label.
5. For Columns Per Row, identify the number of icons to be displayed per row. A value of 5 would display 5 icons per row.
 6. Click **Apply Changes**.

8.4.2.11.2 Adding a Custom Icon View Link

To add a custom Icon View link:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Icon View.
3. For Use Custom Icon Link, select **Yes**.
4. For Columns Per Row, identify the number of icons to be displayed per row.

A value of 5 would display 5 icons per row.

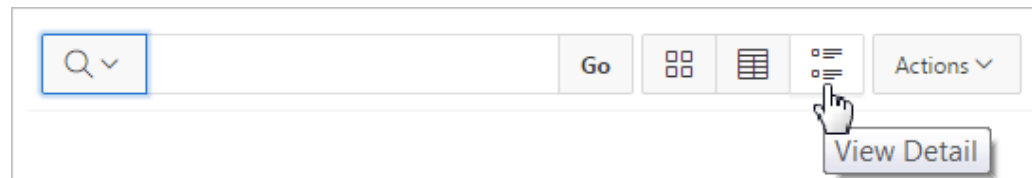
5. In Custom Link, enter the column link for the Icon view. Use substitution strings for column names, for example:

```
<a href="#ICON_LINK#"><br />#PRODUCT_NAME#</a>
```

6. Click **Apply Changes**.

8.4.2.12 Enabling Detail View in Component 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. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Detail View.
3. For Detail View Enabled, select **Yes**.
4. Specify the following attributes:

- Before Rows - Enter the HTML to be displayed before report rows. For example:

```
<TABLE>
```

- For Each Row (Use #COLUMN_ALIAS# substitutions) - Use substitution strings for column names and column labels. For example:


```
<tr><td align="right">#ENAME_LABEL#</td><td>#ENAME#</td></tr>
<tr><td align="right">#JOB_LABEL#</td><td>#JOB#</td></tr>
```
- After Rows - Enter the HTML to be displayed after report rows. For example:


```
</TABLE>
```

5. Click **Apply Changes**.

8.4.2.13 Configuring Interactive Report Advanced Attributes

This section describes how to configure Advanced Attributes section of the Interactive Report Attributes page to define the Interactive Report region alias, specify an item whose value stores the saved report ID, and define a from address in an interactive report subscription or email download.

- [Configuring Advanced Attributes](#) (page 8-84)
- [About Emailing from an Interactive Report](#) (page 8-85)

8.4.2.13.1 Configuring Advanced Attributes

To configure Advanced Attributes:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. Scroll down to Advanced Attributes.

[Table 8-10](#) (page 8-84) describes the available Advanced Attributes.

Table 8-10 Advanced Attributes

Attribute	Description
Region Alias	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 in an API call. For example, you can write a function to return INTERACTIVE_REPORT_ID by querying ALIAS column in the APEX_APPLICATION_PAGE_IR view). The alias must be unique within the application. Since the INTERACTIVE_REPORT_ID value can change when you export or import to different instance, referencing an interactive report region using an alias can be useful.
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.

Table 8-10 (Cont.) Advanced Attributes

Attribute	Description
Email From Address	<p>Enter a valid email address to use as the from address when sending email from an email download or a subscription. The value can be a literal string containing a valid email or an application substitution defined at the application-level using substitution syntax or the <code>&APP_EMAIL.</code> substitution which returns the Application Email From Address defined at the application-level. Examples:</p> <pre>john.doe@abc.com &MY_APP_EMAIL_FROM. &APP_EMAIL.</pre> <p>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.</p> <p>Tip: The email from address comes from the following in the order of precedence: Email From Address at report level > Application Email From Address at application level > Default Email From Address at instance level.</p> <p>If all of them are null, the email from address is the same as email to address.</p> <p>See Also: "Edit Application Definition Page (page 6-17)"</p>

3. Click **Apply Changes**.

8.4.2.13.2 About Emailing from an Interactive Report

You can define the from address for emails sent from an interactive report in two ways:

- At the application-level on the Edit Application Definition page in the Application Email from Address attribute. See "[Edit Application Definition Page](#) (page 6-17)."
- At the interactive report-level on the Interactive Report Attributes page in the Email from Address attribute. See "[Configuring Interactive Report Advanced Attributes](#) (page 8-84)."

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.
```

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:

"[Enabling Email Download](#) (page 8-75)" and "[Understanding Substitution Strings](#) (page 2-26)

8.4.2.14 Editing Saved Default Interactive Reports in Component View

This section describes how developers can view and manage default interactive reports.

- [Editing the Alias of Default Report in Component View](#) (page 8-86)
- [Deleting Saved Default Reports in Component View](#) (page 8-86)



See Also:

"[Saving a Default Interactive Report](#) (page 8-99)" and "Managing Saved Interactive Reports" in *Oracle Application Express Administration Guide*

8.4.2.14.1 Editing the Alias of Default Report in Component View

Developers can edit the report alias or view link examples for default reports on the Saved Reports page.

To edit the report alias for a default report.

1. Navigate to the appropriate page containing the saved interactive report. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Regions, click the **Saved Report** link.
The Saved Reports page appears.
3. In Report Alias, you can either:
 - Enter a new alias for the saved report
 - Copy and paste the link example to link to a specific saved report
4. Click **Apply Changes**.

8.4.2.14.2 Deleting Saved Default Reports in Component View

To delete saved default reports:

1. Navigate to the appropriate page containing the saved interactive report. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Regions, click the **Saved Report** link.
The Saved Reports page appears.
3. To delete a report, select it and click **Delete Checked**.

8.4.2.15 Viewing Saved Interactive Report Activity

Developers can view saved report activity by clicking the **View Saved Report Activity** link under Tasks on the Edit Report Attributes page. This link only displays if saved reports other than the primary default report exist. You can use this page to determine which reports are being used and which ones are not.

To view saved report activity:

1. Access the Interactive Report Attributes page. See "[Accessing the Interactive Report Attributes Page in Component View](#) (page 8-65)."

The Interactive Report Attributes page appears.

2. On the Tasks list, click **View Saved Report Activity**.

Tip:

You can also 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**.

8.4.3 Linking to Interactive Reports

Unlike classic reports which typically use page items as report parameters, the data in an interactive report is driven by declarative filters. However, the developer may want to set these report parameters by modifying interactive report settings for an end user. These settings only affect the end user's working report and do not affect any saved report settings. Use the syntax described in this section in the `Request` value and `ItemNames` section of the URL 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](#) (page 8-88)
- [Name and Value Syntax](#) (page 8-88)
- [ClearCache Syntax](#) (page 8-89)
- [Restrictions When Linking to Interactive Reports](#) (page 8-89)
- [Valid Linking Examples](#) (page 8-90)
- [Invalid Linking Examples](#) (page 8-90)
- [Linking to Shared Interactive Reports](#) (page 8-91)

 **See Also:**

"[Understanding URL Syntax](#) (page 2-16)" and "[About Using f?p Syntax to Link Pages](#) (page 2-17)"

8.4.3.1 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.

8.4.3.2 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`)

8.4.3.3 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.

8.4.3.4 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.

8.4.3.5 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:
```

8.4.3.6 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
```

8.4.3.7 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](#) (page 8-91)
- [Link Examples to a Primary Report](#) (page 8-92)
- [Link Examples to Saved Reports](#) (page 8-92)

See Also:

and

- ["About Using f?p Syntax to Link Pages](#) (page 2-17)"
- ["Saving a Default Interactive Report](#) (page 8-99)"
- ["Deleting Saved Interactive Reports](#) (page 6-81)"
- ["Managing Saved Interactive Reports"](#) in *Oracle Application Express Administration Guide*

8.4.3.7.1 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.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

3. Expand **Identification**.
4. 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.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.4.3.7.2 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
```

8.4.3.7.3 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
```

8.4.4 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.

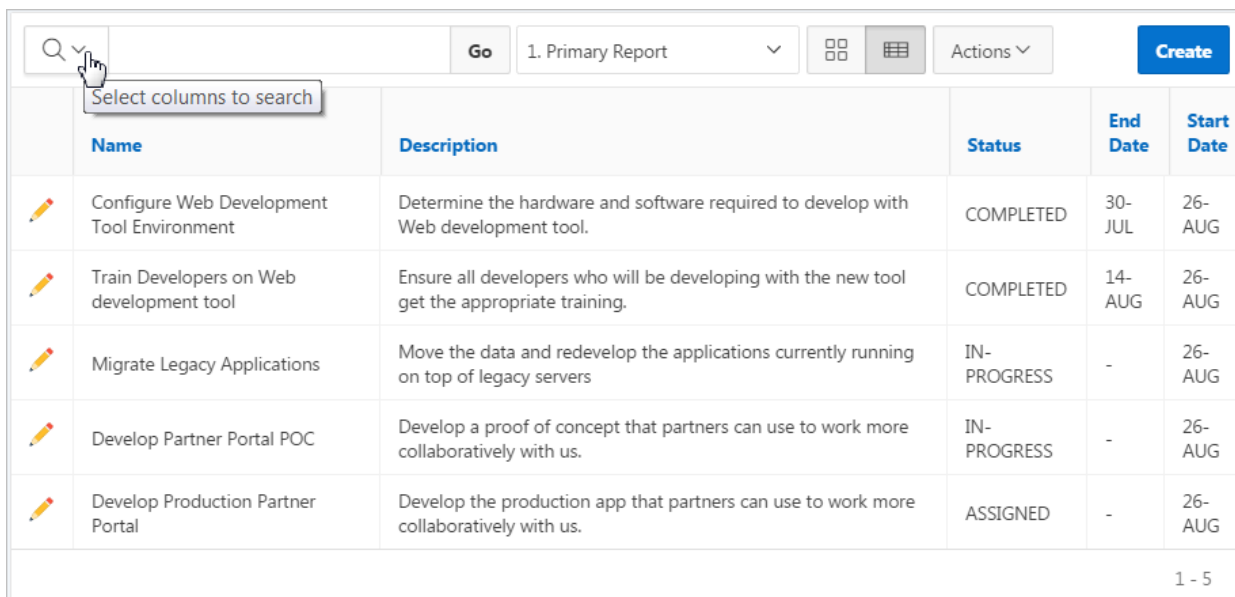
- [About the Search Bar](#) (page 8-93)
- [About the Actions Menu](#) (page 8-94)
- [Using Interactive Report Filters](#) (page 8-95)






- [Saving Interactive Reports](#) (page 8-96)

See Also:

"[Creating Reports Using a Wizard](#) (page 8-5)" and "[Customizing an Interactive Report Using the Actions Menu](#)" in *Oracle Application Express End User Guide*

8.4.4.1 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](#) (page 8-96)."
- **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](#) (page 8-94)."

 **See Also:**

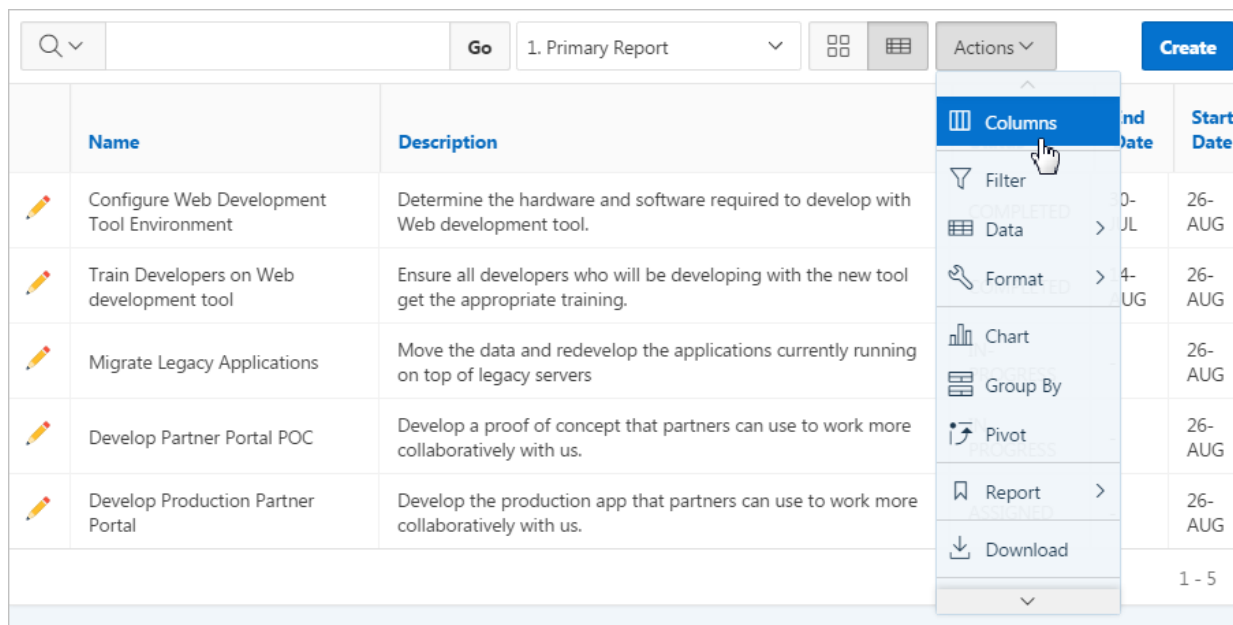
"Customizing the Interactive Report Search Bar (page 8-43)"






8.4.4.2 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 \(page 16-11\)](#)".

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*

8.4.4.3 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

8.4.4.4 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](#) (page 8-96)
- [About Configuration Dependencies](#) (page 8-97)
- [Saving a Public or Private Interactive Report](#) (page 8-98)
- [Saving a Default Interactive Report](#) (page 8-99)
- [Renaming a Public or Private Interactive Report](#) (page 8-100)
- [Deleting a Public or Private Interactive Report](#) (page 8-101)
- [About Exporting Interactive Reports](#) (page 8-101)



See Also:

"Managing Saved Interactive Reports" in *Oracle Application Express Administration Guide*

8.4.4.4.1 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**.

8.4.4.4.2 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 \(page 19-45\)"](#)
- ["Customizing the Actions Menu in Page Designer \(page 8-47\)"](#)
- ["Customizing the Interactive Report Search Bar \(page 8-43\)"](#)

8.4.4.4.3 Saving a Public or Private Interactive Report

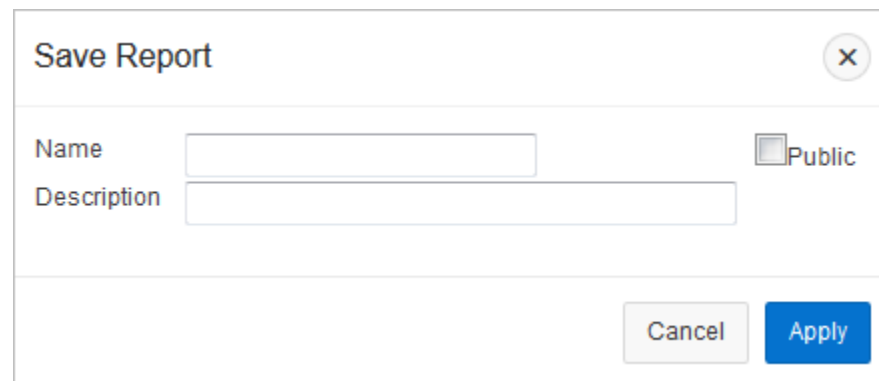
End users can save an interactive report and classify it as being either public or private. Public reports can be viewed by all users. Private reports can only be viewed by the user who saves the report.

 **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 \(page 8-97\)"](#).

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 image shows a 'Save Report' dialog box with 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 screenshot shows a 'Save Report' dialog box. It has a title bar with the text 'Save Report' and a close button (X). Below the title bar, there is a 'Save' dropdown menu currently set to 'As Named Report'. To the right of the dropdown is a note: '(Only displayed for developers)'. Below the dropdown is a 'Name' text input field. Below the 'Name' field is a 'Description' text input field. To the right of the 'Name' and 'Description' fields is a 'Public' checkbox. 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**.

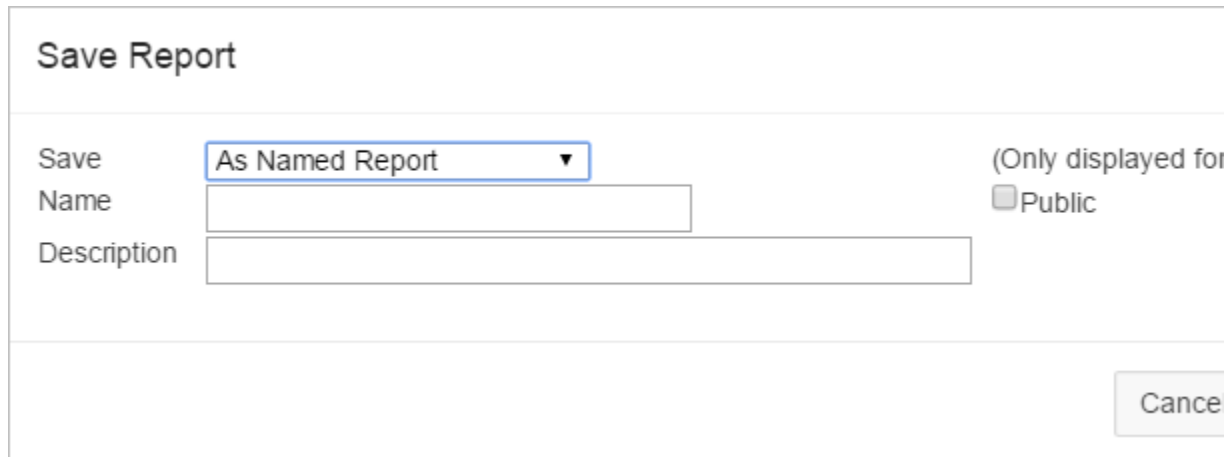
8.4.4.4 Saving a Default Interactive Report

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.

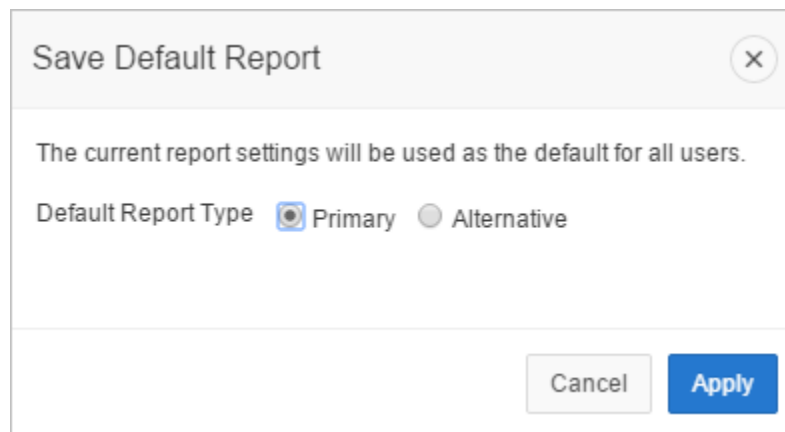


The 'Save Report' dialog box contains the following elements:

- Save**: A dropdown menu currently set to 'As Named Report'.
- Name**: A text input field.
- Description**: A larger text input field.
- (Only displayed for)**: A label next to a checkbox labeled 'Public'.
- Cancel**: A button in the bottom right corner.

4. From Save, select **As Default Report Settings**.

The Save Default Report dialog appears.



The 'Save Default Report' dialog box contains the following elements:

- Save Default Report**: Title bar with a close button (X).
- The current report settings will be used as the default for all users.**: A message.
- Default Report Type**: Radio buttons for 'Primary' (selected) and 'Alternative'.
- Cancel** and **Apply**: Buttons at the bottom.

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**.

8.4.4.4.5 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.

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
Develop Production Partner Portal	Develop the production app that partners can use to work more collaboratively with us.	ASSIGNED	-	26-AUG

- In the Rename Report dialog, edit the attributes (for example, enter a new name) and click **Apply**.

8.4.4.4.6 Deleting a Public or Private Interactive Report

To delete a public or private interactive report:

- Run the report as a developer.
- 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

- Click **Apply**.

8.4.4.4.7 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 (page 23-19)"

8.5 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. With the exception of sorting and simple filtering, end users cannot customize a classic report.

- [Editing Classic Reports in Page Designer](#) (page 8-102)
- [Editing Classic Reports in Legacy Component View](#) (page 8-112)

8.5.1 Editing Classic Reports in Page Designer

Using the Report Attributes page, developers can control report layout, pagination, column sorting, error messages, export links, and column break of Classic reports.

- [Altering Classic Report Layout in Page Designer](#) (page 8-102)
- [Editing Classic Report Attributes in Page Designer](#) (page 8-103)
- [Editing Classic Report Pagination in Page Designer](#) (page 8-104)
- [Enabling Classic Report CSV Output Option in Page Designer](#) (page 8-105)
- [Adding Support for XML File or a CSV File Export in Page Designer](#) (page 8-106)
- [Enabling Column Sorting in a Classic Report in Page Designer](#) (page 8-107)
- [Creating a Column Link in a Classic Report in Page Designer](#) (page 8-107)
- [Adding a Download Link to a Classic Report in Page Designer](#) (page 8-108)
- [Defining an Updatable Column in a Classic Report in Page Designer](#) (page 8-109)
- [Controlling When Classic Report Columns Display in Page Designer](#) (page 8-109)
- [Defining a Column as a List of Values in a Classic Report in Page Designer](#) (page 8-110)
- [Controlling Classic Report Column Breaks in Page Designer](#) (page 8-111)



See Also:

["Managing Pages in Page Designer](#) (page 7-1)" and ["APEX_IR"](#) in *Oracle Application Express API Reference*

8.5.1.1 Altering Classic Report Layout in Page Designer

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.
The Property Editor displays the attributes for that column.
4. Edit the appropriate attributes.
Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
The following table describes common report column edits.

Table 8-11 Common Report Column Edits for Classic Reports

Description	Developer Action
Hide a column.	Locate and expand, Identification . For Type, select Hidden Column .
Edit the column heading text.	Locate and expand Heading . Edit the Heading and Alignment attributes.
Change the column heading text alignment.	Locate and expand Heading . Under Heading, make a new selection from Alignment .
Alter the column display sequence.	Locate and expand Layout . Edit the Sequence attribute.
Enable a unique sort sequence.	Locate and expand 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.	Locate and expand 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.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.5.1.2 Editing Classic Report Attributes in Page Designer

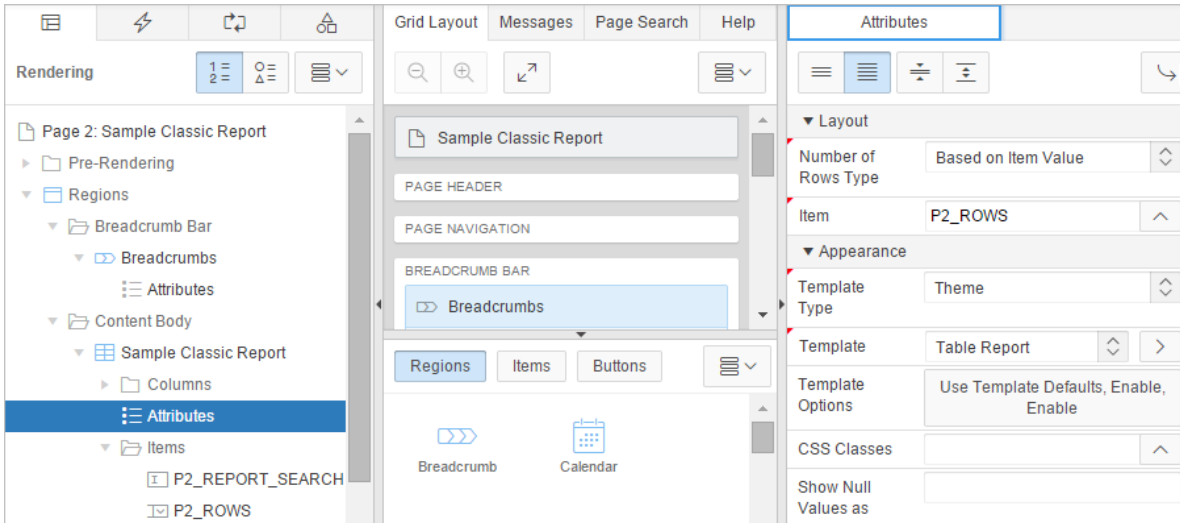
You can customize a classic report by editing report attributes in Page Designer.

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 Region, select the **Attributes** node.



The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

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**.

8.5.1.3 Editing Classic Report Pagination in Page Designer

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 report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Locate and expand **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**.

8.5.1.4 Enabling Classic Report CSV Output Option in Page Designer

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 **Region**, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Locate and expand **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**.

8.5.1.5 Adding Support for XML File or a CSV File Export in Page Designer

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 Region, select the **Attributes** node.

The Property Editor displays the report attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Locate and expand **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**.

8.5.1.6 Enabling Column Sorting in a Classic Report in Page Designer

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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
4. Locate and expand **Sorting**.
5. 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.

6. 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 "[Controlling Classic Report Pagination](#) (page 8-115)."

8.5.1.7 Creating a Column Link in a Classic Report in Page Designer

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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
4. Locate and expand **Identification**. From Type, select **Link**.
5. Locate and expand **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. Reset 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 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**.

8.5.1.8 Adding a Download Link to a Classic Report in Page Designer

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](#) (page 6-32)" and "[Automatic CSV Encoding](#) (page 6-35)"

8.5.1.9 Defining an Updatable Column in a Classic Report in Page Designer

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:
 - a. Locate the region containing the report.
 - b. Expand the Columns node.
 - c. Select the column to be updated.
3. In the Property Editor, expand **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.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.5.1.10 Controlling When Classic Report Columns Display in Page Designer

You can use the Authorization and Condition attributes to control when a column displays.

Authorization enables you 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.

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 and 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 and select the region containing the report.
The region attributes display in the Property Editor.
3. If controlling a column with an authorization scheme, select an authorization scheme:
 - a. Locate and expand **Security**.
 - b. Authorization Scheme - Make a selection.
4. If controlling the column with a condition define a condition, define a condition:
 - a. Locate and expand **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.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

 **See Also:**

- ["Providing Security Through Authorization \(page 19-59\)"](#)
- ["Understanding Conditional Rendering and Processing \(page 2-10\)"](#)
- ["Available Conditions \(page C-1\)"](#)

8.5.1.11 Defining a Column as a List of Values in a Classic Report in Page Designer

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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
4. Locate and expand **Identification**. From Type, select **Plain Text (based on List of Values)**.
5. Locate and expand **List of Values**. Select a **Type** and fill in the appropriate fields.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.5.1.12 Controlling Classic Report Column Breaks in Page Designer

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 Region, select the **Attributes** node.
The Property Editor displays the report attributes in the Property Editor.
Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
4. Specify Break Formatting:
 - a. Locate and expand **Break Formatting**.
 - b. For 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**.

8.5.2 Editing Classic Reports in Legacy Component View

Developers can create and edit classic reports in Component View. Using the Report Attributes page, developers can control report layout, pagination, column sorting, error messages, export links, and column break of Classic reports.

- [Editing Classic Report Attributes in Component View](#) (page 8-112)
- [Altering Classic Report Layout in Component View](#) (page 8-113)
- [Controlling Classic Report Pagination in Component View](#) (page 8-114)
- [Enabling Column Sorting in a Classic Report in Component View](#) (page 8-117)
- [Adding a Download Link to a Classic Report in Component View](#) (page 8-118)
- [Enabling Classic Report CSV Output in Component View](#) (page 8-118)
- [Adding Support for XML File or a CSV File Export](#) (page 8-118)
- [Creating a Column Link in a Classic Report in Component View](#) (page 8-119)
- [Defining an Updatable Column in Component View](#) (page 8-120)
- [Defining a Column as a List of Values in a Classic Report in Component View](#) (page 8-121)
- [Controlling When Classic Report Columns Display in Component View](#) (page 8-122)
- [Controlling Classic Report Column Breaks in Component View](#) (page 8-123)
- [Synchronizing Classic Report Column Order with Region Source](#) (page 8-123)



See Also:

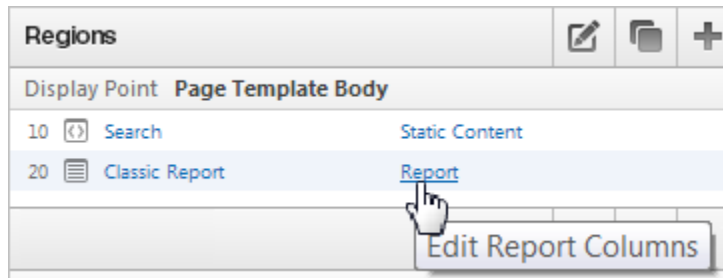
"[Managing Pages in Legacy Component View](#) (page 24-1)," "[Customizing Interactive Reports in a Running Application](#) (page 8-92)," and "APEX_IR" in *Oracle Application Express API Reference*

8.5.2.1 Editing Classic Report Attributes in Component View

This section describes how to customize a classic report by editing report attributes.

To access the Report Attributes page:

1. View the page containing the report in Component View. "[Viewing a Page in Legacy Component View](#) (page 24-10)."
Component View appears.
2. To access the Report Attributes page, click the **Report** link next to the report region you want to edit.



The Report Attributes page appears and is divided into the following sections:

- **Column Attributes** control the report layout. See "[Altering Classic Report Layout in Component View](#) (page 8-113)".
 - **Layout and Pagination** attributes control report pagination. See "[Controlling Classic Report Pagination in Component View](#) (page 8-114)".
 - **Messages** contain attributes that enable you to define messages that display if no data is found or more data is found than can be displayed.
 - **Report Export** attributes enable you to add download link to a report or export a report as either an XML file or CSV file. See "[Adding a Download Link to a Classic Report in Component View](#) (page 8-118)" and "[Adding Support for XML File or a CSV File Export](#) (page 8-118)".
 - **Break Formatting** attributes enable you to control if a specific column repeats and how column breaks appear when printed. See "[Controlling Classic Report Column Breaks in Component View](#) (page 8-123)".
 - **External Processing** attributes enable you to specify a URL for post processing and a link label.
3. To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."
 4. To save your changes, click **Apply Changes**.

8.5.2.2 Altering Classic Report Layout in Component View

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 access the Column Attributes:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears with the Column Attributes section at the top of the page.

Use the Column Attributes section to control report column appearance and functionality.

Heading Type identifies how the heading is generated for the report. The **Link** column indicates if a column link is currently defined. The **Edit** column indicates whether a column is currently updatable.

[Table 8-12](#) (page 8-114) describes common report column edits.

Table 8-12 Common Report Column Edits for Classic Reports

Description	Developer Action
Alter column display sequence.	Click the up and down arrows to change the column display sequence.
Alter heading alignment.	Under Column Alignment, select a column alignment.
Change column heading text.	Under Heading, enter different heading text.
Control which columns display.	Click Show to indicate a column should display.
Enable a unique sort sequence.	Click Sort and select a sequence number from Sort 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.	Click Sum to enable the sum of a column.

You can further refine the attributes of a specific column on the Column Attributes page.

- To access the Column Attributes page, click the **Edit** icon next to the appropriate column Alias.
- To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

See Also:

"[Creating a Column Link in a Classic Report in Component View](#) (page 8-119)," "[Defining an Updatable Column in Component View](#) (page 8-120)," "[Defining a Column as a List of Values in a Classic Report in Component View](#) (page 8-121)," and "[Controlling When Classic Report Columns Display in Component View](#) (page 8-122)"

8.5.2.3 Controlling Classic Report Pagination in Component View

This section describes how to control pagination in a classic report.

- [Pagination in a Classic Report](#) (page 8-114)
- [Controlling Classic Report Pagination](#) (page 8-115)
- [Including Pagination After the Rows in a Classic Report](#) (page 8-116)
- [Including Pagination Before the Rows in a Classic Report](#) (page 8-117)

8.5.2.3.1 Pagination in a Classic Report

For a classic report, you control report pagination by:

- Including a pagination substitution string in the report template

- Making selections from the Layout and Pagination section on the Report Attributes page

You control how pagination displays by making selections from the Layout and Pagination attributes on the Report Attributes page.

8.5.2.3.2 Controlling Classic Report Pagination

To control classic report pagination:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

2. Under Layout and Pagination, select the following:
 - a. Report Template - Select a report template (optional).
 - b. Pagination Scheme - Select a pagination scheme.
 - c. Display Position - Select a display position.
 - d. Number of Rows - Specify how many rows display on each page.
 - e. Click **Apply Changes**.
3. In the Property Editor, locate and edit the following attributes described in [Table 8-13](#) (page 8-115).

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

Table 8-13 Layout and Pagination Attributes

Attribute	Description
Report Template	Specifies a template to be applied to this report. Report templates provide control over the results of a row from your SQL query. You can choose from many default templates, or pick a custom build template.
Report Attributes	Enter a value to replace the #REPORT_ATTRIBUTES# substitution string. The #REPORT_ATTRIBUTES# substitution string is only replaced in the before rows attribute of the report template.
Show Null Values as	Enter the text you want to display for null columns. The default value is (null).
Pagination Scheme	Specifies the type of pagination for this 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 used to navigate to the next or previous page.
Enable Partial Page Refresh	If enabled, this attribute refreshed the report region when paginating forward and backward in your result set.
Display Position	Defines where pagination occurs. If you choose to display pagination above a report, the selected report template must support that type of display.
Number of Rows	Defines the maximum number of rows to display on each page.

Table 8-13 (Cont.) Layout and Pagination Attributes

Attribute	Description
Number of Rows (Item)	Defines the number of rows displayed by default per page for SQL queries (obtained dynamically from an item). Identify the item in this attribute.
Maximum Row Count	<p>Defines the maximum number of rows to query, for example, rows 1 - 10 of 456.</p> <p>If you set this attribute to 200, the result would appear as follows:</p> <pre>rows 1 - 10 of more than 200 rows</pre> <p>This attribute impacts performance. Counting fewer rows can improve performance and counting thousands of rows can degrade performance.</p>
Strip HTML	<p>Specify whether to remove HTML tags from the original column values for HTML expressions, column links and report data exported as CSV files.</p> <p>If you select values from the database that contain HTML tags, then those tags can cause conflicts with the HTML generated for your columns links or HTML expressions. When this option is enabled, only the actual data portion of your column value is used.</p>
Sort Nulls	<p>For reports with column heading sorting, specify if you want null valued columns to sort first or last.</p> <p>See Also: "Enabling Column Sorting in a Classic Report in Component View (page 8-117)"</p>

8.5.2.3.3 Including Pagination After the Rows in a Classic Report

To include pagination after the rows in a report:

1. Create a report. See "[Creating a Report Using the Create Page Wizard](#) (page 8-11)."
2. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

3. Under Layout and Pagination, specify the following:
 - a. Report Template - Select a report template (optional).
 - b. Pagination Scheme - Select a pagination scheme.
 - c. Display Position - Select a display position.
 - d. Number of Rows - Specify how many rows display on each page.
 - e. Click **Apply Changes**.
4. Edit the report template:
 - a. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
 - b. Under Templates, select the report template name.
 - c. Include the #PAGINATION# substitution string in the After Rows attribute.

- d. Click **Apply Changes**.
5. Run the page.

8.5.2.3.4 Including Pagination Before the Rows in a Classic Report

To include pagination before the rows in a report:

1. Create a report. See "[Creating a Report Using the Create Page Wizard](#) (page 8-11)".
2. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

3. Under Layout and Pagination:
 - a. Report Template - Select a report template (optional).
 - b. Pagination Scheme - Select a pagination scheme.
 - c. Display Position - Select a position that contains the word top.
 - d. Number of Rows - Specify how many rows display on each page.
 - e. Click **Apply Changes**.
4. Edit the report template.
 - a. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
 - b. Under Templates, select the report template name.
 - c. Include the #TOP_PAGINATION# substitution string in the Before Rows attribute.
 - d. Click **Apply Changes**.
5. Run the page.

8.5.2.4 Enabling Column Sorting in a Classic Report in Component View

You enable column sorting on the Report Attributes page.

To enable column sorting:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

2. Under Column Attributes, select the **Sort** check box next to the columns to be sorted.
3. From Sort Sequence, select a sequence number.

Sort Sequence is optional. However, if there are one or more sort enabled columns, then at least one column needs a defined Sort Sequence.
4. Click **Apply Changes**.

**Tip:**

Column sorting must be enabled if you want columns with null values to display at the top or end of the report.

8.5.2.5 Adding a Download Link to a Classic Report in Component View

You 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 the CSV output option. When using the CSV output option, 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:**

["Automatic CSV Encoding \(page 6-35\)"](#)

8.5.2.6 Enabling Classic Report CSV Output in Component View

To use the enable the CSV output option:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View \(page 8-112\)](#)."

The Report Attributes page appears.

2. Scroll down to Report Export.
3. From Enable CSV output, select **Yes**.
4. (Optional) In the Separator and Enclosed By fields, define the separator and delimiter.

The default Enclosed By by characters are a double quotation marks (" "). The default delimiter is either a comma or a semicolon depending upon your current NLS settings.

5. In Link Label, enter link text. This text displays in your report and enables users to invoke a download.
6. (Optional) To specify a default export file name, enter a name in the Filename field.

By default, the Application Express engine creates an export file name by taking the region name and adding the appropriate file name extension (.csv or .xml).

7. Click **Apply Changes**.

8.5.2.7 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. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

2. Scroll down to Layout and Pagination.
3. From the Report Template list, select **export: XML** or **export: CSV**.

Selecting **export: XML** prevents the Application Express engine from rendering the page and dumps the content to an XML file.

4. Click **Apply Changes**.

8.5.2.8 Creating a Column Link in a Classic Report in Component View

This section describes how to create a column link to another page in your application or to a URL.

- [Creating a Column Link to Another Page](#) (page 8-119)
- [Creating a Column Link to a URL](#) (page 8-120)

8.5.2.8.1 Creating a Column Link to Another Page

To create a column link to another page:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

2. Under Column Attributes, locate the column to contain the link.
3. Click the **Edit** icon next to the column name.

The Column Attributes page appears.

4. Scroll down to Column Link.
5. To create a column link to another page:
 - a. From Target, select **Page in this Application**.
 - b. (Optional) In Link Attributes, specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
 - c. In Link Text, enter the text to be displayed as a link, specify an image tag, or pick from the list of default images.
 - d. In Page, specify the target page number. To reset the pagination for this page, select **Reset Pagination**.
 - e. In Request, specify the request to be used.
 - f. In Clear Cache, specify the pages (that is, the page numbers) on which to clear cache. You can specify multiple pages by listing the page numbers in a comma-delimited list. See "[Linking to Interactive Reports](#) (page 8-87)."
 - g. Use the Name and Value fields to specify session state for a specific item.
6. Click **Apply Changes**.

8.5.2.8.2 Creating a Column Link to a URL

To create a column link to a URL:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."
The Report Attributes page appears.
2. Access the Column Attributes page by clicking the **Edit** icon next to the appropriate column.
The Column Attributes page appears.
3. Scroll down to Column Link.
4. Under Column Link, specify the following:
 - a. From Target, select **URL**.
 - b. In Link Text, enter the text to be displayed as a link and select a substitution string.
 - c. (Optional) In Link Attributes, specify additional column link attributes to be included in the `` tag (for example, a link target, classes, or styles).
 - d. In URL, enter the appropriate address.
5. Click **Apply Changes**.

8.5.2.9 Defining an Updatable Column in Component View

You can make a column updatable by editing Tabular Form Element attributes on the Column Attributes page. 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. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."
The Report Attributes page appears.
2. Access the Column Attributes page by clicking the **Edit** icon next to the appropriate column.
The Column Attributes page appears.
3. Under Column Attributes, specify the following:
 - a. Display As - Select a type of updatable column.
Use this option to make a column updatable. Updates can only be performed if a multirow update is defined, or a PL/SQL process is implemented to process updated data.
 - b. Number/Date Format - Make a selection if you selected the Display As type of **Date Picker**.

- c. Element Width - Specify the width of the form item.
 - d. Number of Rows - Specify the height of a form item (applicable to text areas).
 - e. Element Attributes - Define a style or standard form element attribute.
 - f. Element Option Attributes - Specify form element attributes for items in a radio group or check box.
 4. Tabular Form Attributes, specify the following:
 - a. Default Type - Identify the default type.
 - b. Default - Identify the default source.
 - c. Reference Table Owner - Identify the owner of the referenced table. This attribute compares references to columns across applications against User Interface Defaults. This reference is manually maintained after being initialized by a wizard, whenever possible.
 - d. Reference Table Name - Specify the table or view that this column references. This attribute compares references to columns across applications against User Interface Defaults. This reference is manually maintained after being initialized by a wizard, whenever possible.
 - e. Reference Column Name - Specify the column name that this report column references. This attribute compares references to columns across applications against User Interface Defaults. This reference is manually maintained after being initialized by a wizard, whenever possible.
 5. Click **Apply Changes**.

8.5.2.10 Defining a Column as a List of Values in a Classic Report in Component View

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. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."
The Report Attributes page appears.
2. Access the Column Attributes page by clicking the **Edit** icon next to the appropriate column.
The Column Attributes page appears. Next, specify how the column is rendered (radio group, LOV, select list, and so on).
3. Under Column Attributes, make a selection from the Display As list.
4. Scroll down to List of Values.
5. From Named LOV, make a selection from the List of Values repository.
6. To include a null value in a list of values:

- a. In Display Null - Select **Yes**.
 - b. Null display value - Enter the display value for the `NULL` option here.
 - c. Null return value - Enter the `NULL` value to be used for this item's list of values. If you do not specify a value, the default is `%null%`.
- A column can also have a value that does not display in its list of values.
7. To define a value that does not display in the list of values:
 - a. From Display Extra Value, select **Yes**.

The extra value is used if the actual column value is not part of the LOV. In that situation, the actual value is shown. If you do not display extra values, you may end up with the wrong value and unintentionally update your data incorrectly.
 - b. In Null Value, specify the value that displays.
 - c. If you have not selected a Named LOV, enter the query used to display a select list in the LOV Query field.
 8. If you have not selected a Named LOV, enter the query used to display a select list in List of values definition.
 9. Click **Apply Changes**.

8.5.2.11 Controlling When Classic Report Columns Display in Component View

You can use the Authorization and Condition attributes to control when a column displays.

Authorization enables you 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.

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 and Condition attributes:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.
2. Access the Column Attributes page by clicking the **Edit** icon next to the appropriate column.

The Column Attributes page appears.
3. If controlling a column with an authorization scheme, make a selection from the Authorization Scheme list.
4. If controlling the column with a condition, define a condition, make a selection from the Condition Type list. Depending upon your selection, enter an expression or value in the appropriate Expression fields.

If the authorization is successful and the condition type display evaluates to true, the column displays.

 **See Also:**

"[Providing Security Through Authorization](#) (page 19-59)," "[Understanding Conditional Rendering and Processing](#) (page 2-10)," and [Available Conditions](#) (page C-1)

8.5.2.12 Controlling Classic Report Column Breaks in Component View

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 the department number only appears once.

To create this type of column break:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

2. Scroll down to Break Formatting.
3. Make a selection from the Breaks list.

8.5.2.13 Synchronizing Classic Report Column Order with Region Source

To synchronize a classic report's column order with the column order defined in the region source query:

1. Access the Classic Report Attributes page. See "[Editing Classic Report Attributes in Component View](#) (page 8-112)."

The Report Attributes page appears.

2. From Tasks, select **Synchronize Column Order with Region Source**.
3. Click Synchronize.

8.6 Printing Report Regions

 **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](#) (page 19-14)."

You can configure a report region to print by exporting it to several different formats. 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.

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.

- [About Printing Reports to PDF](#) (page 8-124)
- [About Report Printing Configuration Options](#) (page 8-124)
- [About Classic Report Printing Methods](#) (page 8-125)
- [Configuring Classic Report Region Print Attributes](#) (page 8-126)
- [Configuring Interactive Report Region Print Attributes](#) (page 8-128)
- [About Report Queries](#) (page 8-130)
- [About Report Layouts](#) (page 8-130)

8.6.1 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.

8.6.2 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*.

8.6.3 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](#) (page 8-126)."
- **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](#) (page 16-46)."

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 \(page 16-49\)"](#)

8.6.4 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 in Page Designer \(page 8-126\)](#)
- [Configuring a Classic Report Print Attributes in Component View \(page 8-127\)](#)

 **See Also:**

["About Classic Report Printing Methods \(page 8-125\)"](#) and ["Configuring Report Printing"](#) in *Oracle Application Express Administration Guide*

8.6.4.1 Configuring a Classic Report Print Attributes in Page Designer

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 in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

4. Enable Printing:
 - a. In the Property Editor, locate and expand **Printing**.
 - b. For **Enabled**, select **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**.

8.6.4.2 Configuring a Classic Report Print Attributes in Component View


To configure a classic report region for printing:

1. In your application, select the page containing the interactive report region you want to print.

The page appears.
2. Under Regions, click **Print** link next to the region you want to print.

The Print Attributes page appears.
3. Under Printing, specify the appropriate information:
 - a. Enable Report Printing - Select **Yes**.
 - b. Link Label - Enter the text for the link that starts the printing process.
 - c. Response Header - Specify whether the response header generated by Oracle Application Express is based on your **Report Settings** or the **Print Server**. The response header is the information the server sends back to the web browser in response to receiving an HTTP request. It contains the date, size and type of file that the server is sending back to the client and also data about the server itself. To specify content disposition (using the attribute View File As) and file name, the response header must be based on Report Settings.
 - 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. (Optional) File Name - Enter a name for the downloaded file. If you leave this blank, the region name is used as the file name.
 - f. Output Format - Select an output option.
 - g. Item - Only enabled, if you select the Output Format **Derived from Item**. If enabled, select the application or page item that holds format value.

- 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 report attributes in the Property Editor.
Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
4. Expand the **Attributes** node and 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**.

8.6.5.2 Configuring Interactive Report Regions for Printing in Legacy Component View

To configure an interactive report region for printing:

1. In your application, select the page containing the interactive report region you want to print.
The page appears.
2. Under Regions, click **Interactive Report** next to the region you want to print.
3. Click the **Print Attributes** tab.
4. Under Printing, specify the appropriate information:
 - a. Response Header - Specify whether the response header generated by Oracle Application Express is based on your Report Settings or the Print Server. The response header is the information the server sends back to the web browser in response to receiving an HTTP request. It contains the date, size and type of file that the server is sending back to the client and also data about the server itself. To specify content disposition (using the attribute View File As) and file name, the response header must be based on Report Settings.
 - b. 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.
 - c. Print Server Overwrite - Specify an alternate print server. A print server is typically configured centrally for all workspaces in Oracle Application Express

Administration Services. See "Configuring Report Printing" in *Oracle Application Express Administration Guide*.

Configuring an alternative print server at the report level allows for easier debugging and diagnosis of potential printing problems and supports the use of specific custom built end points to perform custom XSLT transformations.

5. In the remaining sections, define page size, paper orientation, page headers and footers, fonts, text color, and background color.
6. Click **Apply Changes**.

8.6.6 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 \(page 16-46\)"](#)

8.6.7 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 \(page 16-49\)"](#)

8.7 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 \(page 8-131\)](#)
- [About Providing a Download Link \(page 8-131\)](#)

- [About Displaying the BLOB](#) (page 8-132)
- [About Working With BLOBs Procedurally](#) (page 8-133)

8.7.1 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.

8.7.2 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 <code>BLOB</code> .

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.

8.7.3 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 <code>IMAGE</code> report format mask.
2	Table Name	Yes	Case sensitive name of table containing target column of type <code>BLOB</code> .
3	Column containing BLOB	Yes	Case sensitive name of column of type <code>BLOB</code> .
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 <code>IMAGE</code> format but left in so that the format can easily be changed between <code>IMAGE</code> and <code>DOWNLOAD</code> .

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.

8.7.4 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*

9

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.

Topics:

- [Understanding Form Types](#) (page 9-1)
- [Creating a Form](#) (page 9-5)
- [About Making an Interactive Grid Editable](#) (page 9-14)
- [Managing Master Detail Forms](#) (page 9-14)
- [Validating User Input in Forms](#) (page 9-22)
- [Understanding BLOB Support in Forms](#) (page 9-24)



See Also:

- ["Creating a Legacy Master Detail Form](#) (page 24-53)"
- ["Managing Legacy Tabular Forms](#) (page 24-56)"
- ["Developing Reports](#) (page 8-1)"

9.1 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](#) (page 9-1)
- [Form on Table](#) (page 9-2)
- [Master Detail Forms](#) (page 9-3)



See Also:

- [Creating a Legacy Master Detail Form](#) (page 24-53)

9.1.1 Editable Interactive Grid

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.

The following is an example of an editable interactive grid.

	Lead	Name	Description	Status	Project Start	Completed
<input checked="" type="checkbox"/>	13	Configure Web Development To...	Determine the hardware and softwar...	COMPLETED	26-AUG-16	30-JUL-16
<input type="checkbox"/>	1	Train Developers on Web develo...	Ensure all developers who will be dev...	COMPLETED	26-AUG-16	14-AUG-16
<input type="checkbox"/>	10	Migrate Legacy Applications	Move the data and redevelop the ap...	IN-PROGRE...	26-AUG-16	-
<input type="checkbox"/>	13	Develop Partner Portal POC	Develop a proof of concept that part...	IN-PROGRE...	26-AUG-16	-
<input type="checkbox"/>	1	Develop Production Partner Portal	Develop the production app that par...	ASSIGNED	26-AUG-16	-

See Also:

"Using an Editable Interactive Grid" in *Oracle Application Express End User's Guide*

9.1.2 Form on 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.

The following is an example of a form on table.


Form on a Table - EBA_DEMO_PROJECTS


Lead


* Name

Description

* Status

Completed 

* Created 

* Updated 

 **See Also:**

[Creating a Form Using the Create Page Wizard](#) (page 9-8)

9.1.3 Master Detail Forms

A master detail form reflects a one-to-many relationship between two tables in a database. Typically, a master detail form displays a master row and multiple detail rows within a single HTML form. With this form, users can insert, update, and delete values from two tables or views. In Oracle Application Express, developers can create a single page or two page master detail.

Single Page Master Detail

A single page master detail features two editable interactive grids based on two related tables or views. Users select a row on the master grid to update the detail grid. In this example, the user selects a department on the master grid to view employees in the detail grid. Developers can create a single page master detail with either the Create Application Wizard or the Create Page Wizard.

Departments
Save

<input type="checkbox"/>		Department	Location	Notes
<input checked="" type="checkbox"/>		ACCOUNTING	NEW YORK	-
<input type="checkbox"/>		RESEARCH	DALLAS	-
<input type="checkbox"/>		SALES	CHICAGO	-
<input type="checkbox"/>		OPERATIONS	BOSTON	-

1 rows selected
Total 4

<input type="checkbox"/>		Employee	Job	Manager	Hire Date	Salary	On Leave
<input checked="" type="checkbox"/>		KING	PRESIDENT	-	81-NOV-17	5000	N
<input type="checkbox"/>		CLARK	MANAGER	7839	81-JUN-09	2450	N
<input type="checkbox"/>		MILLER	CLERK	7782	82-JAN-23	1300	N

1 rows selected
Total 3

Two Page Master Detail

A two page master detail features two pages based on two related tables or views. The first page is a non-editable interactive grid. Users click the Edit icon on the first page edit access details on the second page. The second page includes both a form and editable interactive grid. Developers can create a two page master detail with Create Page Wizard.

Two Page Master Detail - Departments
Create

		Department	Location	Notes
		ACCOUNTING	NEW YORK	-
		OPERATIONS	BOSTON	-
		RESEARCH	DALLAS	-
		SALES	CHICAGO	-

1 rows selected
Total 4

In this example, users click the Edit icon to view all employees within a department. The second page includes a form with details about the department and editable interactive grid of employees.

Department

Location

Notes

1 of 4

Cancel
Delete
Save
>

Two Page Master Detail - Employees

		Emp ID	Employee	Job	Manager	Hiredate	Salary	Notes
<input checked="" type="checkbox"/>	☰	7839	KING	PRESIDENT	-	81-NOV-17	5000	-
<input type="checkbox"/>	☰	7782	CLARK	MANAGER	7839	81-JUN-09	2450	-
<input type="checkbox"/>	☰	7934	MILLER	CLERK	7782	82-JAN-23	1300	-

9.2 Creating a Form

Developers can create forms using a number of wizards including the Create Application Wizard, Create Page Wizard, and Create Region Wizard. Note that the available forms differ depending the selected application user interface (for example, Desktop or Mobile).

Tip:

In some wizards you select the table and in others you provide the SQL query. For assistance in writing the SQL query, click the Query Builder button. See "Using Query Builder" in *Oracle Application Express SQL Workshop Guide*.

- [Form Options When Running the Create Page Wizard](#) (page 9-6)
- [Creating a Form Using the Create Page Wizard](#) (page 9-8)
- [Creating a Form Using the Create Application Wizard](#) (page 9-9)

- [Creating a Form Manually in Legacy Component View](#) (page 9-11)
- [Creating a Form Region in Page Designer](#) (page 9-10)

 **See Also:**

- ["Developing Reports](#) (page 8-1)"
- ["Managing Legacy Tabular Forms](#) (page 24-56)"
- ["Creating a Legacy Master Detail Form](#) (page 24-53)"

9.2.1 Form Options When Running the Create Page Wizard

When you create an application, you choose a target user interface. The user interface you select determines the available forms.

Available Forms for Desktop Applications

If you are creating a desktop application, the Create Page Wizard supports the following forms.

Table 9-1 Create Page Wizard — Forms for Desktop Applications

Form Type	Description	To Learn More
Report with Form on Table	Creates two pages. One page displays as an interactive grid. Each row provides a link to the second page to enable users to update each record. You can select the table on which to build the report and form. Note: This wizard does not support tables having more than 127 columns. Selecting more than 127 columns generates an error.	Not applicable.
Editable Interactive Grid	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. You choose a table on which to build the interactive grid.	See " Managing Interactive Grids (page 8-13)" and " Making an Existing Interactive Grid Editable (page 8-21)"
Single Page Master Detail	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.	See " Managing Master Detail Forms (page 9-14)"

Table 9-1 (Cont.) Create Page Wizard — Forms for Desktop Applications

Form Type	Description	To Learn More
Two Page Master Detail	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.	See " Managing Master Detail Forms (page 9-14)"
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 (page 17-20)"
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 (page 17-17)"

Available Forms for Mobile Applications

If you are creating a mobile application, the Create Page Wizard supports the following forms.

Table 9-2 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

Table 9-2 (Cont.) Create Page Wizard — Forms for Mobile Applications

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 (page 17-20)"
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 (page 17-17)"

**See Also:**

["Creating a Form Using the Create Page Wizard \(page 9-8\)"](#)

9.2.2 Creating a Form Using the Create Page Wizard

The easiest way to create a form is to use the Create Page Wizard.

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.
 - b. Select a page type - Select **Form**.
 - c. Click **Next**.
5. Select the type of form page you wish to create. See "[Form Options When Running the Create Page Wizard](#) (page 9-6)."
6. Follow the on-screen instructions.

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

9.2.3 Creating a Form Using the Create Application Wizard

Use the Create Application Wizard to create a form to update a single row in a table. This wizard enables you to choose the table on which to build the 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. For the application type, select **Desktop** and click **Next**.
4. For Name:
 - a. Schema - Each application obtains its privileges by parsing all SQL as a specific database schema. Identify the database schema owner.
 - b. Name - Enter the name used to identify the application to developers.
 - c. Application - Specify an integer value that identifies the application. The application ID must be unique within your Application Express instance. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - d. Theme - Select a theme. Themes are collections of templates used to define the layout and style of an entire application.
 - e. Theme Style - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51.

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.
 - f. Click **Next**.

 **Tip:**

To create an application at this point that consists of just a login page and a blank page, click **Create Application**.

Next, add pages to your application.

5. For Pages, click **Add Page**. In Create Application dialog:
 - a. Select Page Type - Select the page type **Form**.
 - b. Parent Page - If applicable, select a subordinate page. To learn more about an attribute, see field-level Help.
 - c. Table Name - Select a table.
 - d. Click **Add Page**.

The page (or pages) appear at the top of the page. To delete a page, click the **Delete** icon.
6. Repeat the previous steps until all pages have been added.
7. Click **Next**.
8. Follow the on-screen instructions.



Tip:

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



See Also:

- ["Creating a Database Application \(page 6-4\)"](#)
- ["Using Themes \(page 11-1\)"](#)
- ["Using Theme Styles \(page 11-3\)"](#)

9.2.5 Creating a Form Region in Page Designer



Note:

You can also create a form manually by selecting it in the Gallery and dragging and dropping it in the Layout tab. However, Oracle does not recommend a manual approach when creating forms since it does not create all the necessary underlying form items and processes.

To create a form 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 **Form Region**.

The Create Region Wizard appears.

4. Select the type of form you wish to create and click **Next**.

Options include:

- Form on a Table
- Form on a SQL Query
- Form on a Procedure
- Form on a Web Service

To learn more about form region options, see ["Region Types When Running the Create Region Wizard \(page 12-13\)."](#)

5. Follow the on-screen instructions.

9.2.4 Creating a Form Manually in Legacy Component View

This section describes how to create a form manually in Component View.

- [About Creating Forms Manually in Component View](#) (page 9-11)
- [Processing a Form](#) (page 9-11)
- [Populating Forms in Component View](#) (page 9-13)



See Also:

["Managing Pages in Legacy Component View](#) (page 24-1)"

9.2.4.1 About Creating Forms Manually in Component View

You can also create a form manually by performing the following steps:

- Step 1 - Create an HTML region (to serve as a container for your page items)
- Step 2 - Create items to display in the region
- Step 3 - Create processes and branches

To create a form manually by creating an HTML region:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Create an HTML region. See "[Creating a Region in Component View](#) (page 12-18)."
3. Start adding items to the page. "[Understanding Page-Level Items](#) (page 13-1)."

9.2.4.2 Processing a Form

Once you create a form, the next step is to process the data a user types by inserting into or updating the underlying database tables or views. This section describes three ways to process a form.

- [Creating an Automatic Row \(DML\) Processing Process](#) (page 9-11)
- [Creating a Process that Contains One or More Insert Statements](#) (page 9-12)
- [Using a PL/SQL API to Process Form Values](#) (page 9-13)

9.2.4.2.1 Creating an Automatic Row (DML) Processing Process

One common way to implement a form is to manually create an Automatic Row Processing (DML) process. This approach offers three advantages. First, you are not required to provide any SQL coding. Second, Oracle Application Express performs DML processing for you. Third, this process automatically performs lost update detection. Lost update detection ensures data integrity in applications where data can be accessed concurrently.

To implement this approach you must:

- Add items, define the Item Source Type as Database Column, and specify a case-sensitive column name.
- Select the option **Always overrides the cache value**.

To create an Automatic Row Processing (DML) process:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)"
2. Access the Create Page Process Wizard. Under Processes, click the **Create** icon.
3. For Process Type, select **Data Manipulation** and click **Next**.
4. Select the process category **Automatic Row Processing (DML)** and click **Next**.
5. For Process Attributes:
 - a. Name - Enter a name to identify the process.
 - b. Sequence - Specify a sequence number.
 - c. Point - Choose the point of execution for this process (for example, **Onload - After Header**).
 - d. Type - Displays the selected process type.
 - e. Click **Next**.
6. Follow the on-screen instructions.

9.2.4.2.2 Creating a Process that Contains One or More Insert Statements

In this approach to form handling, you create one or more processes to handle insert, update, and delete actions. Instead of having the Application Express engine handling everything transparently, you are in complete control.

For example, suppose you have a form with three items:

- P1_ID - A hidden item to store the primary key of the currently displayed row in a table.
- P1_FIRST_NAME - A text field for user input.
- P1_LAST_NAME - A text field for user input.

Assume also there are three buttons labeled Insert, Update, and Delete. Also assume you have a table T that contains the columns `id`, `first_name`, and `last_name`. The table has a trigger that automatically populates the `ID` column when there is no value supplied.

To process the insertion of a new row, you create a conditional process of type PL/SQL that executes when the user clicks the Insert button. For example:

```
BEGIN
  INSERT INTO T ( first_name, last_name )
    VALUES ( :P1_FIRST_NAME, :P1_LAST_NAME );
END;
```

To process the updating of a row, you create another conditional process of type PL/SQL. For example:

```
BEGIN
  UPDATE T
    SET first_name = :P1_FIRST_NAME,
```



```

        last_name = :P1_LAST_NAME
    WHERE ID = :P1_ID;
END;
```

To process the deletion of a row, you create a conditional process that executes when the user clicks the Delete button. For example:

```

BEGIN
    DELETE FROM T
    WHERE ID = :P1_ID;
END;
```

9.2.4.2.3 Using a PL/SQL API to Process Form Values

For certain types of applications, it is appropriate to centralize all access to tables in a single or a few PL/SQL packages. If you created a package to handle DML operations, you can call procedures and functions within this package from an After Submit PL/SQL process to process insert, updates, and delete requests.

9.2.4.3 Populating Forms in Component View

Oracle Application Express populates a form either on load or when the Application Express engine renders the page. You can populate a form in the following ways:

- Create a process and define the type as Automated Row Fetch.
- Populate the form manually by referencing a hidden session state item.

To create an Automated Row Fetch process:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Access the Create Page Process Wizard. Under Processes, click the **Create** icon.
3. For process type, select **Data Manipulation** and click **Next**.
4. Select the process category, select **Automatic Row Fetch** and click **Next**.
5. For Process Attributes:
 - a. Name - Enter a name to identify the process.
 - b. Sequence - Specify a sequence number.
 - c. Point - Choose the point of execution for this process.
 - d. Type - Displays the selected process type.
 - e. Click **Next**.
6. Follow the on-screen instructions.

You can also populate a form manually by referencing a hidden session state item. For example, the following code in an Oracle Application Express process of type PL/SQL would set the values of `ename` and `sal`. The example also demonstrates how to manually populate a form by referencing a hidden session state item named `P2_ID`.

```

FOR C1 in (SELECT ename, sal
FROM emp WHERE ID=:P2_ID)
LOOP
    :P2_ENAME := C1.ename;
    :P2_SAL := C1.sal;
END LOOP;
```

In this example:

- `c1` is an implicit cursor.
- The value of `P2_ID` has been set.
- The process point for this process would be set to execute (or fire) on or before **Onload - Before Regions**.

9.3 About Making an Interactive Grid Editable

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. Developers determine whether the underlying data is read-only or editable by users.

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](#) (page 8-21)" and "[Managing Interactive Grids](#) (page 8-13)"

9.4 Managing Master Detail Forms

A master detail form reflects a one-to-many relationship between two tables in a database. Typically, a master detail form displays a master row and multiple detail rows within a single HTML form. With this form, users can insert, update, and delete values from two tables or views.



Tip:

To see a working examples of one page and two page master detail forms, install the sample application, *Sample Projects* and view the *Projects* page. See "[Installing a Packaged Application](#) (page 3-4)."

- [About Master Detail Forms](#) (page 9-15)

- [Creating a Single Page Master Detail Using the Create Application Wizard \(page 9-18\)](#)
- [Creating a Single Page Master Detail Using the Create Page Wizard \(page 9-19\)](#)
- [Creating a Two Page Master Detail Using the Create Page Wizard \(page 9-21\)](#)



See Also:

["Creating Master Detail from an Existing Interactive Grid \(page 8-15\)"](#) and ["Creating a Legacy Master Detail Form \(page 24-53\)"](#)

9.4.1 About Master Detail Forms

A master detail form reflects a one-to-many relationship between two 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.

Single Page Master Detail

A single page master detail features two editable interactive grids based on two related tables or views. Users select a row in the master grid to update the detail grid. Developers can create a single page master detail with either the Create Application Wizard or the Create Page Wizard.

A single page master detail features two editable interactive grids based on two related tables or views.

Departments
Save

<input type="checkbox"/>	<input type="checkbox"/>	Department	Location	Notes
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ACCOUNTING	NEW YORK	-
<input type="checkbox"/>	<input type="checkbox"/>	RESEARCH	DALLAS	-
<input type="checkbox"/>	<input type="checkbox"/>	SALES	CHICAGO	-
<input type="checkbox"/>	<input type="checkbox"/>	OPERATIONS	BOSTON	-

1 rows selected
Total 4

Employees

<input type="checkbox"/>	<input type="checkbox"/>	Employee	Job	Manager	Hire Date	Salary	On Leave
<input checked="" type="checkbox"/>	<input type="checkbox"/>	KING	PRESIDENT	-	81-NOV-17	5000	N
<input type="checkbox"/>	<input type="checkbox"/>	CLARK	MANAGER	7839	81-JUN-09	2450	N
<input type="checkbox"/>	<input type="checkbox"/>	MILLER	CLERK	7782	82-JAN-23	1300	N

1 rows selected
Total 3

Users select a row on the master grid to update the detail grid. In this example, when the user selects a department on the master grid to view employees in the detail grid. Developers can create a single page master detail with either the Create Application Wizard or the Create Page Wizard

Employees

<input type="checkbox"/>	<input type="checkbox"/>	Emp ID	Employee	Job	Manager	Hire Date	Salary	Notes
<input type="checkbox"/>	<input type="checkbox"/>	7839	KING	PRESIDENT	-	81-NOV-17	5000	-
<input type="checkbox"/>	<input type="checkbox"/>	7782	CLARK	MANAGER	7839	81-JUN-09	2450	-
<input type="checkbox"/>	<input type="checkbox"/>	7934	MILLER	CLERK	7782	82-JAN-23	1300	-

<

>





Total 3

Two Page Master Detail

A two page master detail features two pages based on two related tables or views. The first page is a non-editable interactive grid. Users click the Edit icon on the first page edit details on the second page.

Two Page Master Detail - Departments [Create](#)

Q v Go Actions v [Reset](#)

	Department ↑≡	Location	Notes
	ACCOUNTING	NEW YORK	-
	OPERATIONS	BOSTON	-
	RESEARCH	DALLAS	-
	SALES	CHICAGO	-

1 rows selected Total 4

In this example, users click the Edit icon to access all employees within a department. The second page includes both a standard form (for the master) and editable interactive grid of employees.

Department

Location

Notes

1 of 4

[Cancel](#) [Delete](#) [Save](#) [>](#)

Two Page Master Detail - Employees

Q v Go Actions v Edit Add Row [Reset](#)

<input type="checkbox"/>	≡	Emp ID	Employee	Job	Manager	Hiredate	Salary	Notes
<input checked="" type="checkbox"/>	≡	7839	KING	PRESIDENT	-	81-NOV-17	5000	-
<input type="checkbox"/>	≡	7782	CLARK	MANAGER	7839	81-JUN-09	2450	-
<input type="checkbox"/>	≡	7934	MILLER	CLERK	7782	82-JAN-23	1300	-

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.

9.4.2 Creating a Single Page Master Detail Using the Create Application Wizard

When you create a master detail form using the Create Application Wizard, the wizard creates single page with two editable interactive grids based on two related tables or views. With this page, users can query, insert, update, and delete values from two tables or views which are related.

**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. For the application type, select **Desktop**.
4. For Name:
 - a. Schema - Each application obtains its privileges by parsing all SQL as a specific database schema. Identify the database schema owner.
 - b. Name - Enter the name used to identify the application to developers.
 - c. Application - Specify an integer value that identifies the application. The application ID must be unique within your Application Express instance. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
 - d. Theme - Select a theme. Themes are collections of templates used to define the layout and style of an entire application.
 - e. Theme Style - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51.
 - f. Click **Next**.
5. Click the **Add Page** button.

On Create Application:

- a. Select Page Type - Select **Master Detail**.
 - b. Parent Page - If applicable, select a subordinate page.
 - c. Master Table Name - Select the table or view to be used as your master table. This table is used to create a report page and a master form on the master detail page.
 - d. Detail Table Name - Select the table or view to be used as your detail table.
Only tables and views that are associated to your selected Master Table using a foreign key are displayed.
 - e. Click **Add Page**.
The new page appears at the top of the Create Application page.
 - f. Click **Next**.
6. For Shared Components, indicate whether to copy shared components from another application. Select **Yes** or **No**. If you select **Yes**, select where to copy them from and which components to import.
 7. For Attributes, configure as appropriate and click **Next**. To learn more about an attribute, see field-level Help.
 8. Confirm your selections and click **Create Application**.
 9. To view the new master detail page, click **Run Page**.
The rendered page appears.

 **See Also:**

- ["Using Themes \(page 11-1\)"](#)
- ["Using Theme Styles \(page 11-3\)"](#)
- ["Managing Shared Components \(page 16-1\)"](#)

9.4.3 Creating a Single Page Master Detail Using the Create Page Wizard

Using the Create Page Wizard, you can create either a Single Page Master Detail. A Single Page Master Detail 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. Navigate to the application, or create a new application.
2. On the Workspace home page, click the **App Builder** icon.
3. Select an application.
4. Click **Create Page**.
5. For Create a Page:
 - a. User Interface - Select a user interface for the page.

- 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**.
10. 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.
11. To view the new master detail, click **Save and Run Page**.

The rendered pages appear.

9.5 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](#) (page 9-22)
- [Associating a Validation with a Specific Item in Page Designer](#) (page 9-22)
- [Associating a Validation with a Specific Item in Component View](#) (page 9-23)
- [About the Error Message Display Location](#) (page 9-24)

9.5.1 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](#) (page 15-8)."
- Step 2 - Associate the validation with a specific item.

9.5.2 Associating a Validation with a Specific Item in Page Designer

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. To view all attributes in the Property Editor, click **Show All**.
5. To search for a group or an attribute, enter a keyword in the Filter Properties field.
6. 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.

7. Click **Save**.

9.5.3 Associating a Validation with a Specific Item in Component View

To associate an item with a validation and specify error message text:

1. Navigate to the appropriate page in Component View. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Navigate to the Edit Page Validations page. Under Validations, select the validation you want to associate.
The attributes page for the validation appears.
3. Scroll down to Error Message.
4. In Error Message:
 - Error Message - Enter an error message.
 - Error message display location - Identifies where the validation error message displays. Verify the display location.
 - Associated Item - Select the item you want to associate with this validation.

5. Click **Apply Changes**.

9.5.4 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.

9.6 Understanding BLOB Support in Forms

This section describes BLOB support in forms.

- [About BLOB Support in Forms](#) (page 9-24)
- [About Uploading and Downloading Files into a BLOB](#) (page 9-24)
- [About Displaying the BLOB](#) (page 9-25)
- [About Removing the Image Reference](#) (page 9-25)

9.6.1 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.

9.6.2 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](#) (page A-1)."

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.

9.6.3 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](#) (page A-1)." To handle it procedurally, see "[Understanding BLOB Support in Forms and Reports](#) (page 8-130)."

9.6.4 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;
```

10

Managing Database Application Components

Oracle Application Express supports the creation of calendars, maps, charts, and application Help.

Topics:

- [Creating Calendars](#) (page 10-1)
- [Creating Maps](#) (page 10-12)
- [Creating Charts](#) (page 10-23)
- [Creating Help for Your Application](#) (page 10-33)



See Also:

"[Developing Reports](#) (page 8-1)," "[Controlling Page Layout](#) (page 12-1)," and "[Adding Navigation](#) (page 14-1)"

10.1 Creating Calendars

App Builder includes a built-in wizards for generating 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](#) (page 10-2)
- [Sample Calendar](#) (page 10-3)
- [Adding a Calendar to a New Page](#) (page 10-4)
- [Creating a Calendar in Page Designer](#) (page 10-5)
- [Managing Calendar Attributes in Page Designer](#) (page 10-6)
- [About Dynamic Action Support for Calendar](#) (page 10-11)



See Also:

"[Managing Legacy Calendars](#) (page 24-61)" and "[Integrating with Google Calendar](#) (page 16-10)"

10.1.1 About Supported Calendars

App Builder supports two types of calendars:

- **Calendar** - Calendar is based on the FullCalendar jQuery library and can only be customized through CSS.
- **Legacy Calendar** - 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 options 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 webservice 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](#) (page 24-61)"

10.1.2 Sample Calendar

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 Eugene Bradley [\$870.00]	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 Packaged Application](#) (page 3-4)"

10.1.3 Adding a Calendar to a New Page

Run the Create Application Wizard to add a calendar to a new page.

To create a Calendar from the Application home 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. Click **Create Page**.
The Create Page Wizard appears.
2. For Create a Page:
 - a. User Interface - Select a user interface for the page.
 - b. Select a page type - Select **Calendar**.
 - c. Click **Next**.
3. For Page Attributes:

**Tip:**

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)".

- 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. Breadcrumb - Select whether to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - e. 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 Source, select if the calendar source is a table or SQL query.
If the Calendar source is 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**.

6. 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**.

10.1.4 Creating a Calendar in Page Designer

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 \(page 10-4\)"](#).

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.

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. To view all attributes in the Property Editor, click **Show All**.
6. To search for a group or an attribute, enter a keyword in the Filter Properties field.
7. In the Property Editor, edit the appropriate Region attribute:
 - Identification, Title - Enter a region title.
 - Source, SQL Query - Select the column which holds the CSS Class to style the events displayed on this calendar.

 **Tip:**

To learn more about default classes and view an example, select the SQL Query attribute in the Property Editor and click the **Help** tab in the central pane.

8. 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.
9. Click **Save**.

 **See Also:**

["Managing Calendar Attributes in Page Designer \(page 10-6\)"](#)

10.1.5 Managing Calendar Attributes in Page Designer

Once you create a calendar, you can alter the display by editing attributes in the Properties 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](#) (page 10-7)
- [Editing an Existing Calendar to Include Add and Edit Functionality](#) (page 10-7)
- [Adding Calendar Initialization JavaScript Code](#) (page 10-9)
- [Enabling the Dragging and Dropping of Data In an Existing Calendar](#) (page 10-10)

10.1.5.1 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.
4. To view all attributes in the Property Editor, click **Show All**.
5. To search for a group or an attribute, enter a keyword in the Filter Properties field.
6. Edit the calendar attributes.

Tip:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

7. Click **Save**.

10.1.5.2 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.
4. To view all attributes in the Property Editor, click **Show All**.
5. To search for a group or an attribute, enter a keyword in the Filter Properties field.
6. 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.
7. Edit **Settings, Edit Link** to define a target page to be called when the user clicks an existing entry.

**Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

 **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`.
8. Click **Save**.

10.1.5.3 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.
3. To view all attributes in the Property Editor, click **Show All**.
4. To search for a group or an attribute, enter a keyword in the Filter Properties field.
5. Locate and expand **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.

- 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;
}
```

- 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 Packaged Application](#) (page 3-4)."

10.1.5.4 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:

- View the page in Page Designer:
 - On the Workspace home page, click the **App Builder** icon.
 - Select an application.
 - Select a page.

Page Designer appears.
- In the Rendering tab under the calendar title, select the **Attributes** node.
The Property Editor displays the Attributes for the calendar.
- To view all attributes in the Property Editor, click **Show All**.
- To search for a group or an attribute, enter a keyword in the Filter Properties field.

5. Locate and expand **Settings** and edit the following attributes:
 - a. Settings, Primary Key Column - Select a table or SQL Query column.
Drag and Drop appears under Settings.
 - 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.
6. Click **Save**.

10.1.6 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 the 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 Packaged Application](#) (page 3-4)."

 **See Also:**

"[Managing Dynamic Actions](#) (page 13-36)" and "[Debugging Dynamic Actions](#) (page 13-50)" for information on how to debug problems

10.2 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](#) (page 10-13)

- [About Creating SQL Queries for Maps](#) (page 10-13)
- [Adding Map Charts to a New Page](#) (page 10-16)
- [Managing Map Charts in Page Designer](#) (page 10-17)
- [Managing Map Charts in Legacy Component View](#) (page 10-20)

10.2.1 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/>

10.2.2 About Creating SQL Queries for Maps

You define a map in App Builder using a wizard. You select a map type, map source, and provide a SQL query using the following syntax:

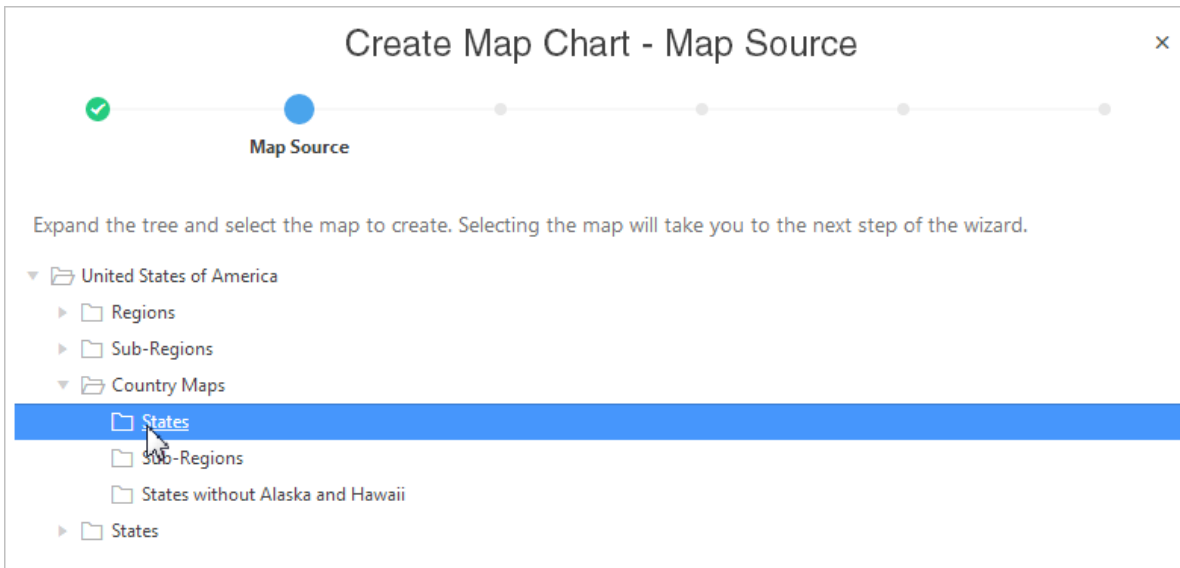
```
SELECT link, label, value  
FROM ...
```

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** and click **Next**.
6. For Map Type, select **United States of America** and click **Next**.
7. For Map Source, expand Country Maps and select **States**.



Create Map Chart

Query

Enter SQL Query or PL/SQL function returning a SQL Query: FROM (SELECT 'Ireland' country, 4203200 people FROM dual)

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	

Specify a SQL query by either:

- Entering a SQL query in the field provided. See "[About Creating SQL Queries for Maps](#) (page 10-13)."
- 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**.

10.2.3 Adding Map Charts to a New Page

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.
 - 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 \(page 10-13\)"](#)

10.2.4 Managing Map Charts in Page Designer

Add or edit map charts in Page Designer.

- [Adding a Map Chart in Page Designer \(page 10-17\)](#)
- [Editing Map Chart Attributes in Page Designer \(page 10-18\)](#)
- [Using Custom XML with Map Charts \(page 10-19\)](#)
- [Enabling Automatic Updates \(page 10-20\)](#)

10.2.4.1 Adding a Map Chart 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**.

10.2.4.2 Editing Map Chart Attributes in Page Designer

Once you create a map chart, you can alter its display by editing attributes in the Property Editor.

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 \(page 10-19\)"](#) and ["Enabling Automatic Updates \(page 10-20\)"](#)

10.2.4.3 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/>

10.2.4.4 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. Locate and expand **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.

5. Click **Save**.

10.2.5 Managing Map Charts in Legacy Component View

Developers can create and edit map charts in Component View.

- [Adding a Map Chart in Component View](#) (page 10-21)
- [Editing Map Chart Attributes](#) (page 10-22)
- [Using Custom XML with Map Charts](#) (page 10-22)
- [Enabling Asynchronous Updates](#) (page 10-23)

**See Also:**

[Managing Pages in Legacy Component View](#) (page 24-1)

10.2.5.1 Adding a Map Chart in Component View

To add a map chart to an existing page.

1. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Create a new region. Under Region, click the **Create** icon.
The Create Region Wizard appears.
3. Select **Map Chart** and click **Next**.
4. For Map Type, select a map type (for example, **Europe**), and click **Next**.
5. For Map Source, a Tree lists the map source files for the selected Map Type. Click the Map Source you want to use (for example, **Denmark**).
6. For Display Attributes:
 - a. Title - Enter a title for the region. This title displays if the region template you choose displays the region title.
 - b. Region Template - Choose a template to control the look of the region.
 - c. Parent Region - Define the parent region to which the new region belongs.
 - d. Display Point - Identify a display point for this region. To learn more, see field-level Help.
 - e. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - f. Click **Next**.
7. For Display Attributes, select the appropriate map attributes and click **Next**.
To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."
8. Specify a query by either:
 - Entering a SQL query in the field provided. See "[About Creating SQL Queries for Maps](#) (page 10-13)."
 - 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 Region 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. See "[About Creating SQL Queries for Maps](#) (page 10-13)."

9. For Conditional Display:

- a. Condition Type - Select a condition type from the list that must be met in order for this component to be rendered processed.
- b. Authorization Scheme - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.
- c. Click **Create Region**.

10.2.5.2 Editing Map Chart Attributes

Once you create a map, you can alter its display by editing map attributes on the Map Attributes page.

To access the Map Attributes page:

1. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Map Attributes**.
3. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)".
4. Click **Apply Changes**.



See Also:

["Using Custom XML with Map Charts](#) (page 10-22)" and ["Enabling Asynchronous Updates](#) (page 10-23)"

10.2.5.3 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. Create a map chart.
2. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
3. Access the Map Attributes page. Under Region, select the region name.
4. Click **Map Attributes**.
5. Scroll down to **Map XML**.
6. From Use Custom XML, select **Yes**.

If you select to use custom XML, regions with attributes that no longer appear are hidden. To display these regions again, select **No** for Use Custom XML.

7. Edit the XML.
8. Click **Apply Changes**.

 **Tip:**

For more information on supported XML format for maps, see the Online Map XML Reference at:

<http://6.anychart.com/products/anymap/docs/>

10.2.5.4 Enabling Asynchronous 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 asynchronous updates:

1. Create a map chart.
2. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
3. Access the Map Attributes page.
4. Under Region, select the region name.
5. Click **Map Attributes**.
6. Scroll down to Refresh.
7. From Asynchronous Update, select **Yes**.
8. In Update Interval (Seconds), enter the interval in seconds between map updates. For optimal performance, select an interval that is greater than 2 seconds.
9. Click **Apply Changes**.

10.3 Creating Charts

App Builder includes built-in wizards for generating charts.

- [About Supported Charts](#) (page 10-24)
- [Sample Charts](#) (page 10-24)
- [Adding a Chart to a New Page](#) (page 10-25)
- [Managing Charts](#) (page 10-27)
- [Using Custom JavaScript with Charts](#) (page 10-32)

**See Also:**

[Managing Legacy Charts](#) (page 24-32)

10.3.1 About Supported Charts

App Builder supports two types of charts:

- **Chart** - Chart support in Oracle Application Express is based on the Oracle JavaScript Extension Toolkit (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*.

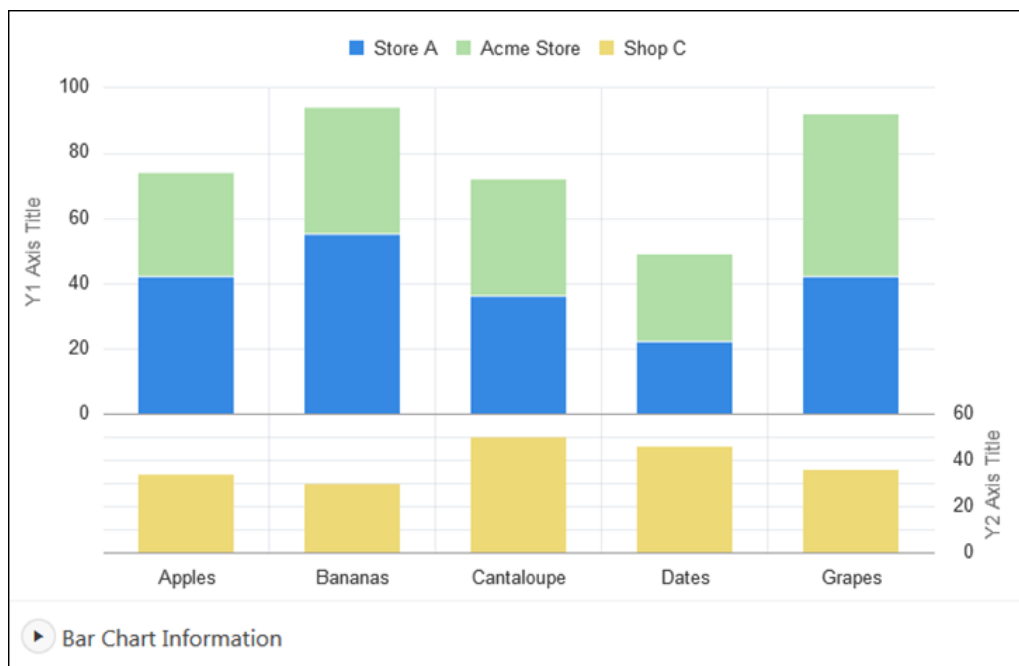
- **AnyChart Chart** - AnyChart chart support is based on a third party charting solution provided by AnyChart. This is a flexible Flash and JavaScript (HTML5) based solution that enables developers to create animated and compact interactive charts. AnyChart charts have been categorized as a legacy component and may be deprecated in the future release.

**See Also:**

"[Managing Legacy Charts](#) (page 24-32)" and "[Upgrading an Application to Include New Components](#) (page 6-76)"

10.3.2 Sample Charts

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 Packaged Application](#) (page 3-4)" for more information on installing the *Sample Charts* application

10.3.3 Adding a Chart to a New Page

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**.
4. For Create a Page:
 - a. User Interface - Select a user interface for the page.
 - b. Select a page type - Select **Chart**.
 - c. Click **Next**.

5. For Chart Type:
 - a. Chart Type - Select a chart type.

**Tip:**

If you select Gantt, you are prompted to select a Gantt chart type.

- b. Click **Next**.
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.
To learn more, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."
 - 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 if the chart source is a table or SQL query.
If the Chart source is table:
 - a. Source Type - Select **Table**.
 - b. Table/View Owner - Select the owner of the table on which you are building a 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 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 will vary depending upon the chart type selected.

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."
 - b. Click **Next**.
 10. Click **Create**.

10.3.4 Managing Charts

Developers can create and edit chart attributes in Page Designer.

- [Adding a Chart in Page Designer](#) (page 10-27)
- [Editing Chart Attributes in Page Designer](#) (page 10-29)
- [Switching Chart Type in Page Designer](#) (page 10-29)
- [Enabling Automatic Refresh](#) (page 10-30)



See Also:

["Viewing a Page in Page Designer](#) (page 7-22)"

10.3.4.1 Adding a Chart 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. 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. Edit the required region attributes.

View all attributes in the Property Editor, select **Show All**.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. Edit the chart attributes:

- a. In the Rendering tab, select the **Attributes** node.

The Property Editor displays **Attributes**.

- b. Edit the required chart attributes.

7. 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.

8. Edit the axes:

 **Tip:**

The Axes node will only be visible 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.

9. Click **Save** or **Save and Run Page**.

10.3.4.2 Editing Chart Attributes in Page Designer

Once you have created a chart, you can alter its display by editing chart attributes in Page Designer. Oracle JET based charts cannot be maintained using the Legacy Component View.

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.
4. To view all attributes in the Property Editor, click **Show All**.
5. To search for a group or an attribute, enter a keyword in the Filter Properties field.
6. Edit the chart attributes.
7. Click **Save**.

10.3.4.3 Switching Chart Type in Page Designer

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. Expand the **Chart** section.
4. For Type, select the chart type you want to switch to. For example, Combination.
To learn more about an attribute, see field-level Help.
5. 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.

6. Click **Save** or **Save and Run Page**.

10.3.4.4 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](#) (page 10-30)
- [Creating a Refresh Dynamic Action in Page Designer](#) (page 10-30)

10.3.4.4.1 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. To view all attributes in the Property Editor, click **Show All** .
4. Locate **Automatic Refresh**.

 **Tip:**

To search for a group or an attribute, enter a keyword in the Filter Properties field.

5. For Automatic Refresh, select **Yes**.
6. Click **Save** .

10.3.4.4.2 Creating a Refresh Dynamic Action in Page Designer

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 \(page 13-36\)"](#)

10.3.5 Using Custom JavaScript with Charts

Control the look and feel of a chart by adding custom JavaScript.

To use custom JavaScript:

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. To view all attributes in the Property Editor, click **Show All**.
4. Locate **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 \(page 24-35\)"](#)

10.4 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 in Page Designer](#) (page 10-33)
- [Creating Help in Legacy Component View](#) (page 10-35)
- [Editing Multiple Item Help Topics Simultaneously](#) (page 10-38)
- [Creating a Help Navigation Bar Entry](#) (page 10-39)

10.4.1 Creating Help in Page Designer

This section describes how to create Help in Page Designer.

- [About Creating Help](#) (page 10-33)
- [Creating a Help Page and Region in Page Designer](#) (page 10-33)
- [Defining Page Help in Page Designer](#) (page 10-35)
- [Defining Item Help in Page Designer](#) (page 10-35)

10.4.1.1 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 in Page Designer](#) (page 10-33)."
- **Step 2:** Define page Help text. See "[Defining Page Help in Page Designer](#) (page 10-35)."
- **Step 3:** Define item Help text. See "[Defining Item Help in Page Designer](#) (page 10-35)."
- **Step 4:** Create a navigation bar icon to link to your Help page. See "[Creating a Help Navigation Bar Entry](#) (page 10-39)."

10.4.1.2 Creating a Help Page and Region in Page Designer

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 view all attributes in the Property Editor, click **Show All** .
5. Edit the following attributes:



Tip:

To search for a group or an attribute, enter a keyword in the Filter Properties field.

- 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:

["Creating a Page from the Application Home Page \(page 6-48\)"](#)

10.4.1.3 Defining Page Help in Page Designer

To define Page Help text in Page Designer:

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. In Property Editor - Page, scroll down to **Help** and enter text in **Help Text** attribute.
4. Click **Save**.

10.4.1.4 Defining Item Help in Page Designer

To define Page Help text in Page Designer:

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. In the Property Editor, locate and expand **Help** and enter text in **Help** attribute,
4. Click **Save**.
5. Repeat the previous steps for each item requiring Help text.

10.4.2 Creating Help in Legacy Component View

This section describes how to create Help in Component View.



Note:

[Managing Pages in Legacy Component View \(page 24-1\)](#)

- [About Creating Help \(page 10-36\)](#)
- [Creating a Help Page and Region in Component View \(page 10-36\)](#)

- [Defining Page Help in Component View](#) (page 10-37)
- [Defining Item Help in Component View](#) (page 10-37)

10.4.2.1 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 in Component View](#) (page 10-36)."
- **Step 2:** Define page Help text. See "[Defining Page Help in Component View](#) (page 10-37)."
- **Step 3:** Define item Help text. See "[Defining Item Help in Component View](#) (page 10-37)."
- **Step 4:** Create a navigation bar icon to link to your Help page. See "[Creating a Help Navigation Bar Entry](#) (page 10-39)."

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.

10.4.2.2 Creating a Help Page and Region in Component View

The first step in creating Help for your application is to create a dedicated page and Help Text region. The region you create displays page and item help for the current page when called using following syntax:

```
f?p=&APP_ID. :Page:Session:&APP_PAGE_ID.
```

To create a Help page and region in Component View:

1. Create a blank page for your Help. See "[Creating a Page from the Application Home Page](#) (page 6-48)."
2. View the page in Component View. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
3. Create a new region. Under Regions, click the **Create** icon.
The Create Region Wizard appears.
4. Under Regions, the **Create** icon.
5. When prompted to select a region type, select **Help Text** and click **Next**.
6. For Display Attributes:
 - a. Title - Enter a title for the region. This title displays if the region template you select displays the region title.
 - b. Region Template - Choose a template to control the appearance of the region.
 - c. Parent Region - Defines the parent region to which the new region belongs.
 - d. Display Point - Identify a display point for this region. To learn more, see field-level Help.
 - e. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.

- f. Click **Next**.
7. On Source, click **Next**.
8. For Conditional Display:
 - a. Condition Type - Select a condition type from the list that must be met in order for this component to be rendered processed. See "[Understanding Conditional Rendering and Processing](#) (page 2-10)."
 - b. Authorization Scheme - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed. See "[Providing Security Through Authorization](#) (page 19-59)."
 - c. Click **Next**.
9. Click **Create Region**.

10.4.2.3 Defining Page Help in Component View

To define page Help text in Component View:

1. Navigate to the page for which you want to add page Help. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Page, click the **Edit** icon.
3. Scroll down to **Help**.
4. Enter your Help text in the field provided.

 **Tip:**

By default, Help is populated with the text `No help is available for this page.`

5. Click **Apply Changes**.

Repeat the previous procedure for each page requiring page Help text.

10.4.2.4 Defining Item Help in Component View

To define item Help in Component View:

1. Navigate to the page for which you want to add Help. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Items, locate the item you want to edit.
3. Access the attributes page for the item. Under Items, select the item name.
The Edit Page Item page appears.
4. Scroll down to **Help Text**.
5. Enter your Help text in the field provided.
6. Click **Apply Change**.

Repeat the previous procedure for each item requiring Help text.

10.4.3 Editing Multiple Item Help Topics Simultaneously

This section explains various ways to edit multiple Help topics at once.

- [Using the Bulk Edit Item Help Report](#) (page 10-38)
- [Seeding Item Help Topics](#) (page 10-38)

10.4.3.1 Using the Bulk Edit Item Help Report

If you are including Help in your application, you can edit all Help topics within an application at once using the Bulk Edit Item Help report. In addition, you 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.

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 \(page 10-38\)"](#)

10.4.3.2 Seeding Item Help Topics

If your application does not yet contain Help, you can perform a mass update 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**.

10.4.4 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 **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. See "[Viewing Field-Level Help](#) (page 1-27)."
 - 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. 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 in Page Designer](#) (page 10-33)."
 - 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**.



See Also:

["Creating a Navigation Bar Entry \(page 14-19\)"](#)

11

Using Themes and Theme Styles

Developers can alter a database application's user interface and page layout through themes and templates.

Topics:

- [Using Themes](#) (page 11-1)
- [Managing Themes and Subscriptions](#) (page 11-17)
- [Creating Custom Themes](#) (page 11-32)
- [Using Custom Cascading Style Sheets](#) (page 11-81)

11.1 Using Themes

Use themes and theme styles to increase your productivity when building an application. Themes are collections of templates that define the layout and style of an application. If a theme includes themes styles, developers can alter the appearance of an application using the Theme Roller Utility.

- [About Themes](#) (page 11-1)
- [About Responsive Design](#) (page 11-2)
- [About Grid-based Themes](#) (page 11-3)
- [Using Theme Styles](#) (page 11-3)
- [Understanding Template Options](#) (page 11-8)
- [Accessing the Themes Page](#) (page 11-9)
- [Editing Themes](#) (page 11-10)

See Also:

["Managing Themes and Subscriptions](#) (page 11-17)" and ["Managing Themes and Subscriptions](#) (page 11-17)"

11.1.1 About Themes

Themes are collections of templates that enable you to define the layout and style of an entire application. The idea behind a theme is to provide 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.

Oracle Application Express ships with an extensive theme repository. 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](#) (page 11-25)."
- **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*.

Newer themes, such as *Universal Theme - 42* and *Mobile - 51*, include 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.



See Also:

["Using Theme Styles \(page 11-3\)"](#)

11.1.2 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 full experience on larger screens. On smart phones and tablets, the layout can also 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. *Universal Theme - 42* is an example of a responsive user interface theme.

Keep in mind, that responsive design is not just a matter of picking one set of templates. As the application developer, you are responsible for using the templates in theme and design a page to facilitate a truly responsive design. Converting an existing application to the Universal Theme requires you review all the application pages to ensure you have defined the appropriate template and layout.

Oracle Application Express provides responsive user interface themes that enable you to build applications that can scale to any screen size. *Universal Theme - 42* is a responsive Desktop theme. Developers can use this theme to build applications that work well on a variety of different screen sizes.

Mobile Theme - 51 enables you to build applications that are optimized for the use on mobile devices such as smartphones and tablets. Pages built with the Mobile theme are smaller and lighter weight than Desktop pages and provide support for touch-events and orientation-change.

**Tip:**

See "[Utilizing Packaged Applications](#) (page 3-1)" to view applications using *Universal Theme - 42*

11.1.3 About Grid-based Themes

Oracle Application Express applications define the position of regions and items on a page using grid settings. Older themes, such as *Theme 26 - Productivity Applications*, use a table-based grid. In table-based themes, regions can be positioned next to each other and forms can have two or more items in a single row. Newer themes, such as *Universal Theme - 42*, use a DIV-based grid. Pages using a DIV-based grid offer the following advantages:

- Easier to maintain and update. DIV-based pages have less code and therefore less complexity.
- Faster loading. Pages using a DIV-based grid generally load faster.

If you switch from a table-based theme to a DIV-based theme, you need to configure the Reset Grid option, **Reset fixed region position** in the Switch Theme Wizard. Selecting **Reset fixed region position** removes existing settings and renders all regions and items stacked vertically.

**See Also:**

"[Switching Themes](#) (page 11-21)"

11.1.4 Using Theme Styles

Newer themes, such as *Universal Theme - 42* and *Mobile - 51*, include theme styles. 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.

- [About Theme Styles](#) (page 11-3)
- [Creating a Theme Style](#) (page 11-4)
- [Editing a Theme Style from the Themes Page](#) (page 11-5)
- [Editing a Theme Style from the User Interface Page](#) (page 11-5)
- [Enabling Users to Select a Theme Style](#) (page 11-5)
- [Using Theme Roller](#) (page 11-6)

11.1.4.1 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* and *Mobile - 51* 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 "[Using the Create Application Wizard](#) (page 6-1)."
- Editing the User Interface. See "[Editing a Theme Style from the User Interface Page](#) (page 11-5)."
- Accessing the Create/Edit Theme page. See "[Editing a Theme](#) (page 11-10)" and "[Creating a Theme Style](#) (page 11-4)."

11.1.4.2 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**.

11.1.4.3 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**.

11.1.4.4 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**.

11.1.4.5 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.

11.1.4.6 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.

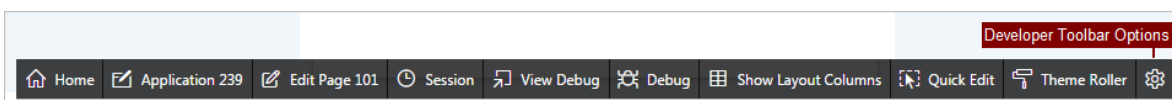
Tip:

Install and run a packaged application that uses the *Universal Theme - 42* to see an example of Theme Roller. See "[Utilizing Packaged Applications](#) (page 3-1)."

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.

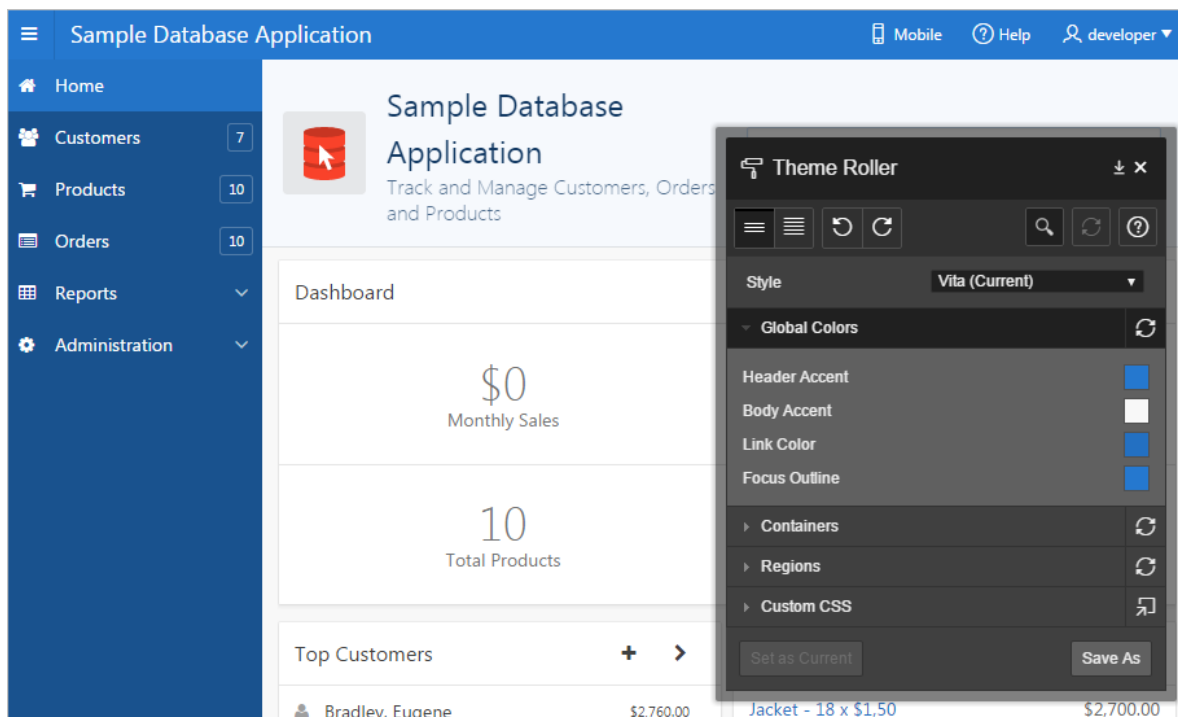


Tip:

Theme Roller only displays in the Runtime Developer Toolbar if theme styles have been defined. See "[Editing a Theme](#) (page 11-10)" and "[Styles](#) (page 11-16)."

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.

11.1.5 Understanding Template Options

Template options provide a declarative way for developers to apply different styles to components on an Oracle Application Express page.

- [About Template Options](#) (page 11-8)
- [About Default Template Options](#) (page 11-9)
- [About Template Option Groups and Presets](#) (page 11-9)

**See Also:**

["Using Template Options](#) (page 11-35)" and ["Using Quick Edit to Modify Live Template Options](#) (page 11-40)"

11.1.5.1 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.

11.1.5.2 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.

11.1.5.3 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 it is created. Template options that are part of a group are optional unless a present is defined at the template-level.

11.1.6 Accessing the Themes Page

You 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](#) (page 11-10)".

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.

11.1.7 Editing Themes

This section describes how to edit a theme attributes.



See Also:

"[Creating Custom Themes](#) (page 11-32)"

- [Editing a Theme](#) (page 11-10)
- [Theme Attributes](#) (page 11-11)

11.1.7.1 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](#) (page 11-11)."
8. To save any changes, click **Apply Changes**.

11.1.7.2 Theme Attributes

This section describes the attributes on the Create/Edit Theme Attributes page. To learn more about an attribute and view examples, see field-level Help.

- [Name](#) (page 11-11)
- [Theme Subscription](#) (page 11-12)
- [JavaScript and Cascading Style Sheets](#) (page 11-12)
- [Component Defaults](#) (page 11-12)
- [Region Defaults](#) (page 11-14)
- [Dialog Defaults](#) (page 11-14)
- [Transition Defaults](#) (page 11-15)
- [Global Templates Options](#) (page 11-15)
- [Icons](#) (page 11-15)
- [Image](#) (page 11-16)
- [Styles](#) (page 11-16)
- [Files](#) (page 11-16)

11.1.7.2.1 Name

[Table 11-1](#) (page 11-11) describes the attributes under Name.

Table 11-1 Name

Attribute	Description
Application	Identifies the current application ID. See Also: " Editing Application Attributes (page 6-16)"
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 (page 6-40)"
Navigation Type	Identifies the selected Navigation Type. Valid options include: Tabs or List. See Also: " Creating Tabs (page 14-2)" and " Creating Lists (page 16-27)"
Navigation Bar Implementation	Identifies the selected Navigation Bar Implementation. Valid options include: Classic or List. See Also: " User Interface Details Page (page 6-40)"
Description	Enter a description for the theme.

11.1.7.2.2 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 \(page 11-17\)"](#)

11.1.7.2.3 JavaScript and Cascading Style Sheets

[Table 11-2](#) (page 11-12) describes the attributes under JavaScript and Cascading Style Sheets.

Table 11-2 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 <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. JavaScript file URLs you enter here replaces the <code>#THEME_JAVASCRIPT#</code> 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 <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.

11.1.7.2.4 Component Defaults

[Table 11-3](#) (page 11-12) describes the default templates by component type.

Table 11-3 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. See Also: "Display Attributes (page 24-14)"

Table 11-3 (Cont.) Component Default Templates

Attribute	Description
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. See Also: " Display Attributes (page 24-14)"
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.
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.

Table 11-3 (Cont.) Component Default Templates

Attribute	Description
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.

11.1.7.2.5 Region Defaults

[Table 11-4](#) (page 11-14) describes the default templates available under Region Defaults.

Table 11-4 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

11.1.7.2.6 Dialog Defaults

[Table 11-5](#) (page 11-14) describes the default templates available under Dialog Defaults.

Table 11-5 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.

Table 11-5 (Cont.) Dialog Defaults

Attribute	Description
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.

11.1.7.2.7 Transition Defaults

Transition Defaults is only visible for a Mobile theme. [Table 11-6](#) (page 11-15) describes attributes available under Transition Defaults.

Table 11-6 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.

11.1.7.2.8 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](#) (page 11-8)" and "[Managing Global Template Options](#) (page 11-37)"

11.1.7.2.9 Icons

[Table 11-7](#) (page 11-15) describes attributes under Icons.

Table 11-7 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.

Table 11-7 (Cont.) Icons

Attribute	Description
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>

11.1.7.2.10 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**.

11.1.7.2.11 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 \(page 11-3\)"](#)

11.1.7.2.12 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](#) (page 16-12)" and "[Managing Static Workspace Files](#) (page 16-17)"

11.2 Managing Themes and Subscriptions

Developers can manage themes directly or through the use of theme subscriptions.

When 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. Additionally, developers cannot modify the theme or template metadata when they access it from the theme or template edit pages. Another advantage of theme subscriptions is that master themes can be upgraded during future Oracle Application Express releases.

- [Understanding Theme Subscriptions](#) (page 11-17)
- [Creating a Theme](#) (page 11-20)
- [Switching Themes](#) (page 11-21)
- [Changing a Theme Identification Number](#) (page 11-23)
- [Copying a Theme](#) (page 11-24)
- [Deleting a Theme](#) (page 11-24)
- [Creating a Master Theme Application](#) (page 11-25)
- [Managing Workspace Themes](#) (page 11-25)
- [Managing Instance Themes](#) (page 11-28)
- [About Exporting and Importing Themes](#) (page 11-28)
- [Viewing Theme Reports](#) (page 11-29)

11.2.1 Understanding Theme Subscriptions

This section describes theme subscriptions.

- [About Theme Subscriptions](#) (page 11-18)
- [About Modifying a Theme Subscription](#) (page 11-18)
- [Verifying if a Theme Refresh Is Needed](#) (page 11-18)
- [Refreshing a Subscribed Theme](#) (page 11-19)
- [Unsubscribing to a Theme](#) (page 11-19)
- [Re-subscribing to a Built-in Theme](#) (page 11-20)

11.2.1.1 About Theme Subscriptions

Developers subscribe to a theme when:

- **Running the Create Application Wizard.**

When a developer select a built-in theme when running the Create Application Wizard, 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 "[Using the Create Application Wizard](#) (page 6-1)."

- **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](#) (page 11-20)."

When an application subscribes to a master theme, 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.

11.2.1.2 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](#) (page 11-24)"

11.2.1.3 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.

11.2.1.4 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](#) (page 11-20)."

8. Click **Apply Changes**.

11.2.1.5 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](#) (page 11-20)."

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**.

11.2.1.6 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](#) (page 11-23)."
3. Run the Create Theme Wizard and create the theme again. See "[Creating a Theme](#) (page 11-20)."
4. Switch the current application to the use the new theme. See "[Switching Themes](#) (page 11-21)."
5. Delete the original theme from which you created a new identification number. See "[Deleting a Theme](#) (page 11-24)."

11.2.2 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 \(page 11-17\)](#)."

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 \(page 11-10\)"](#)

11.2.3 Switching Themes

This section describes how to change the active theme for an application.

- [About Switching the Active Theme \(page 11-21\)](#)
- [About Switching to the Universal Theme \(page 11-22\)](#)
- [Switching an Active Theme \(page 11-22\)](#)

11.2.3.1 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](#) (page 11-20)".
- 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.

11.2.3.2 About Switching to the Universal Theme

When compared to themes provided with previous releases of Oracle Application Express, the Universal Theme has a number of new features, improvements, and optimizations. Additionally, Universal Theme utilizes several key UI features of Oracle Application Express release 5 and provides a number of new templates and template options that greatly expand the ability to customize of your application's UI. To learn more about the advantages of the Universal Theme and how to switch to it see the *Migration Guide - Transitioning to Universal Theme* application:

<https://apex.oracle.com/ut>

11.2.3.3 Switching an Active 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.

Tip:

If you do not have at least two themes installed, click **Create Theme**. You can create a theme from scratch or simply select an existing theme from the repository. See "[Creating a Theme](#) (page 11-20)."

2. Click **Switch Theme**.
The Switch Theme page appears.
3. For Identify Theme:
 - a. Currently Active Theme - Select the current theme for the application.

- b. Switch to Theme - Select the new theme for the application.
- c. Reset Grid - Select whether to reset the grid. Grid settings define the positioning of your regions and items on a page. To learn more, see field-level Help.
- d. Match Template Classes - Select **Yes** or **No**. When switching to a new theme, App Builder maps all currently used templates to a template in the new theme using the template class. When Match Template Class is set to **No**, then templates can be switched to any template with the same type, but irrespective of the template class.
- e. 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**.

11.2.4 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.

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**.

11.2.7 Creating a Master Theme Application

Developers can create a master theme application. As a best practice, Oracle recommends starting with an existing packaged application that uses *Universal Theme - 42* (for example, *Sample Database Application*).

- [Creating a Master Theme Application by Creating a New Application](#) (page 11-25)
- [Creating a Master Theme Application from an Existing Application](#) (page 11-25)

11.2.7.1 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.

11.2.7.2 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](#) (page 11-24)."
2. Switch to the new theme. See "[Switching Themes](#) (page 11-21)."
3. Follow the on-screen instructions.

11.2.8 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](#) (page 11-26)
- [Viewing an Existing Workspace Theme](#) (page 11-27)
- [Deleting a Workspace Theme](#) (page 11-27)
- [Modifying a Workspace Theme](#) (page 11-27)
- [Exporting a Workspace Theme](#) (page 11-28)



See Also:

"Managing Public Themes" in *Oracle Application Express Administration Guide*

11.2.8.1 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**.

11.2.8.2 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](#) (page 11-27)."

11.2.8.3 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**.

11.2.8.4 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 a Database Application](#) (page 6-4)."
2. Modify the theme. See "[Editing Themes](#) (page 11-10)."
3. Delete the existing workspace theme. See "[Deleting a Workspace Theme](#) (page 11-27)."
4. Add the modified theme to the workspace theme repository. See "[Adding a Workspace Theme to the Theme Repository](#) (page 11-26)."

11.2.8.5 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 a Database Application](#) (page 6-4)."
2. Export the application. See "[Exporting an Application](#) (page 23-19)."
3. Import the exported file into the target Oracle Application Express instance. See "[Importing an Application, Page or Component Export](#) (page 23-31)."
4. Install the exported file from the Export Repository. See "[Installing Export Files](#) (page 23-35)."

11.2.9 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*

11.2.10 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 a Database Application](#) (page 6-4)."
2. Export the theme. See "[Exporting Themes](#) (page 23-26)."
3. Import the exported file into the target Oracle Application Express instance. See "[Importing Export Files](#) (page 23-30)."
4. Install the exported file from the Export Repository. See "[Installing Export Files](#) (page 23-35)."

11.2.11 Viewing Theme Reports

This section describes several reports designed to help you manage themes and templates.

- [Viewing All Templates in a Theme](#) (page 11-29)
- [Viewing Theme Template Counts](#) (page 11-29)
- [Viewing File References](#) (page 11-30)
- [Viewing Class References](#) (page 11-30)
- [Viewing Template Substitution Strings](#) (page 11-31)
- [Viewing All Templates in a Theme](#) (page 11-29)
- [Viewing Theme Template Counts](#) (page 11-29)
- [Viewing File References](#) (page 11-30)
- [Viewing Class References](#) (page 11-30)
- [Viewing Template Substitution Strings](#) (page 11-31)

11.2.11.1 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 a theme or select **All**.
 - c. Click **Go**.

A list of templates appears with the template type, template name, the associated theme, and template class.

4. To edit a template, select the template name.

11.2.11.2 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. See "[Accessing the Themes Page](#) (page 11-9)."

2. Click **Reports**.
3. On the Theme Reports page:
 - a. Report - Select **Theme Template Counts**.
 - b. Theme - Select a theme or select **All Themes**.
 - c. Click **Go**.
4. If you are using custom classifications, select **Show Custom** and click **Go**.

11.2.11.3 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.

11.2.11.4 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.

11.2.11.5 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:**

["Understanding Substitution Strings \(page 2-26\)"](#)

11.3 Creating Custom Themes

The Application Express engine creates the user interface of an application based on a named collection of templates called a theme. Templates control the look and feel of the components in your 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 \(page 11-32\)](#)
- [About Calling the JavaScript File from the Page Template \(page 11-33\)](#)
- [About Using Escaping Syntax in Substitution Strings \(page 11-33\)](#)
- [Selecting a Default Page Template \(page 11-34\)](#)
- [Creating a New Template \(page 11-35\)](#)
- [Using Template Options \(page 11-35\)](#)
- [Viewing Template Reports \(page 11-42\)](#)
- [Managing Templates \(page 11-43\)](#)
- [Breadcrumb Templates \(page 11-47\)](#)
- [Button Templates \(page 11-49\)](#)
- [Legacy Calendar Templates \(page 11-51\)](#)
- [Label Templates \(page 11-53\)](#)
- [List Templates \(page 11-56\)](#)
- [Page Templates \(page 11-59\)](#)
- [Popup LOV Templates \(page 11-67\)](#)
- [Region Templates \(page 11-69\)](#)
- [Report Templates \(page 11-72\)](#)

**See Also:**

["Using Themes \(page 11-1\)"](#) and ["Managing Themes and Subscriptions \(page 11-17\)"](#)

11.3.1 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 \(page 11-81\)"](#)

11.3.2 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 \(page 11-61\)"](#)

11.3.3 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 \(page 2-27\)"](#)

11.3.4 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](#) (page 11-34)
- [Selecting a Page-level Template for a Specific Page](#) (page 11-34)

11.3.4.1 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](#) (page 11-10)"

11.3.4.2 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 in Page Designer:
 - 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**.
4. To edit the page-level template in Component View:
 - a. View the page in page in Component View.
 - b. Under Page, click the **Edit** icon.
The Edit page appears.
 - c. Locate **Display Attributes**. From Page Template, select a new template.
 - d. Click **Apply Changes**.

**See Also:**

"[Viewing a Page in Page Designer](#) (page 7-22)" and "[Viewing a Page in Legacy Component View](#) (page 24-10)"

11.3.5 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.

11.3.6 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](#) (page 11-36)
- [Managing Global Template Options](#) (page 11-37)
- [Managing Template Options](#) (page 11-39)



See Also:

["Understanding Template Options](#) (page 11-8)"

11.3.6.1 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](#) (page 11-36)
- [Editing Global Template Option Groups](#) (page 11-37)



See Also:

["About Template Option Groups and Presets](#) (page 11-9)" and ["Understanding Template Options](#) (page 11-8)"

11.3.6.1.1 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**.

11.3.6.1.2 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**.

11.3.6.2 Managing Global Template Options

This section describes how to create and edit global template options.

- [Creating Global Template Options](#) (page 11-37)
- [Editing Global Template Options](#) (page 11-38)



See Also:

["About Template Options](#) (page 11-8)"

11.3.6.2.1 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.

 **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**.

11.3.6.3 Managing Template Options

This section describes how to create and edit template options.

- [Creating Template Options](#) (page 11-39)
- [Editing Template Options](#) (page 11-39)
- [Using Quick Edit to Modify Live Template Options](#) (page 11-40)

 **See Also:**

["Understanding Template Options](#) (page 11-8)"

11.3.6.3.1 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**.

11.3.6.3.2 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**.

11.3.6.3.3 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.

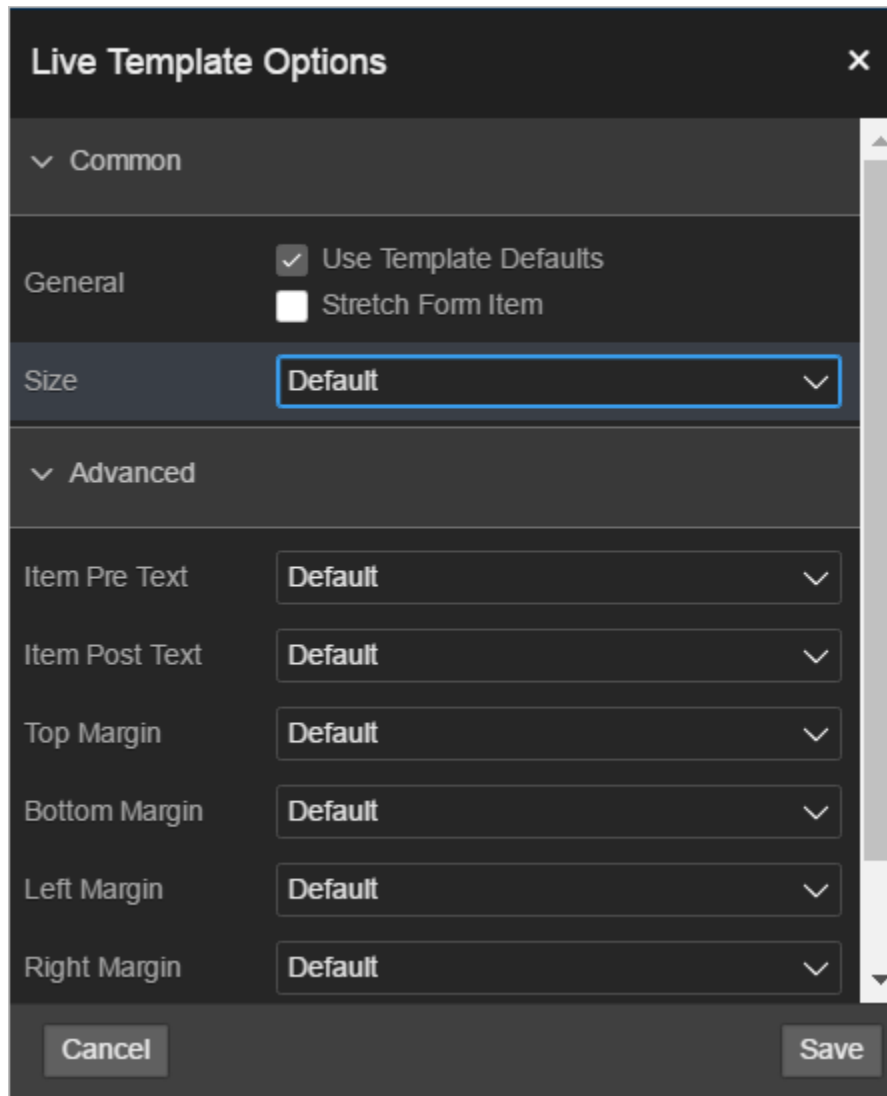
To access the Live Template Options:

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	<input type="text"/>	* Last Name	<input type="text"/>
Street Address	<input type="text"/>	Line 2	<input type="text"/>
City	<input type="text"/>	* State	- Choose State -
* Zip Code	<input type="text"/>		
* Credit Limit	<input type="text"/>		
Phone Number	999-999-9999	Alternate Number	999-999-9999
Email	<input type="text"/>	URL	<input type="text"/>
Tags	<input type="text"/>		

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 \(page 24-29\)"](#)

11.3.7 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.

11.3.8 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](#) (page 11-43)
- [Viewing Templates Associated with a Specific Page](#) (page 11-44)
- [Replacing Templates](#) (page 11-44)
- [Replacing All Templates within an Application](#) (page 11-45)
- [Viewing Region Position Utilization by Page Template](#) (page 11-45)
- [Unsubscribing to Templates](#) (page 11-46)
- [Publishing Templates](#) (page 11-46)

 **See Also:**

["Viewing All Templates in a Theme](#) (page 11-29)"

11.3.8.1 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**.

11.3.8.2 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**.
4. View the page in Component View:
 - a. Under Page, click the **Edit** icon.
The Edit page appears.
 - b. Locate **Display Attributes. Page Template** displays the current template.
 - c. To save any changes, click **Apply Changes**.

Note:

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**.

See Also:

["Viewing a Page in Page Designer \(page 7-22\)"](#) and ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

11.3.8.3 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**.

11.3.8.4 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**.

11.3.8.5 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.

11.3.8.6 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**.

11.3.8.7 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**.

11.3.9 Breadcrumb Templates

A breadcrumb template controls the display of breadcrumb entries. You select a breadcrumb template when you create a region.

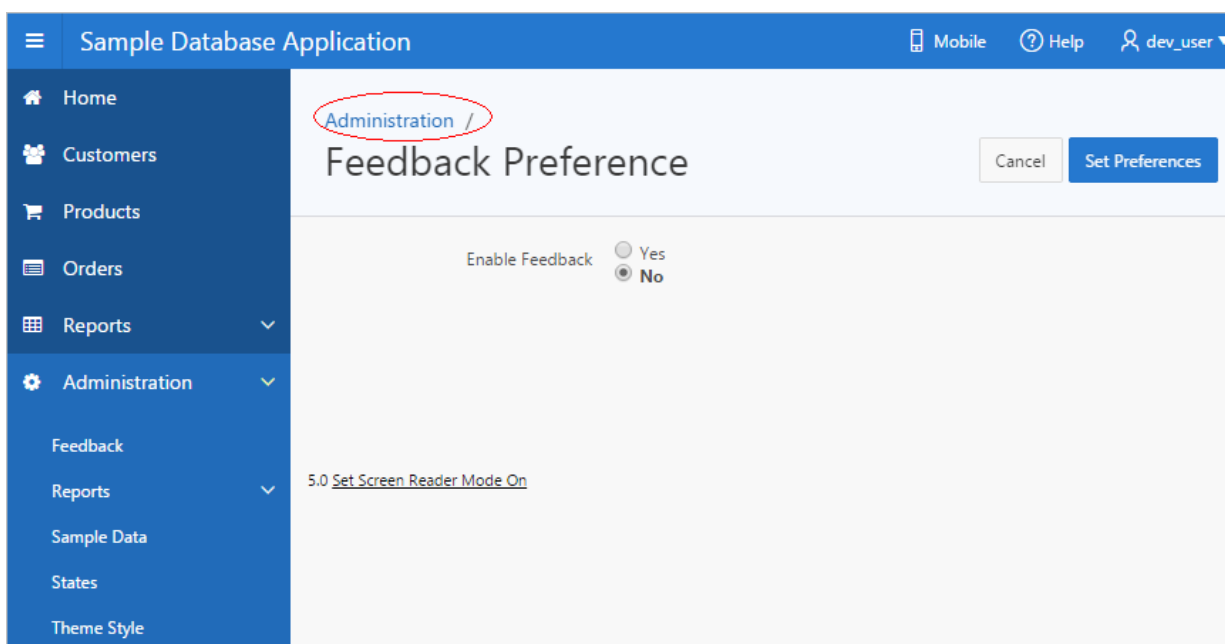
 **See also:**

["Using Themes \(page 11-1\)"](#)

- [About Breadcrumb Style Navigation \(page 11-47\)](#)
- [Breadcrumb Template Attributes \(page 11-48\)](#)

11.3.9.1 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 \(page 14-8\)"](#)

11.3.9.2 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](#) (page 11-48)
- [Definition](#) (page 11-48)
- [Substitution Strings](#) (page 11-48)

11.3.9.2.1 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.

11.3.9.2.2 Definition

Definition attributes control how a breadcrumb displays [Table 11-8](#) (page 11-48) describes Definition attributes.

Table 11-8 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 in Page Designer (page 12-7)" and " Editing Region Attributes in Component View (page 12-19)"
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.

11.3.9.2.3 Substitution Strings

Substitution strings are used within subtemplates to reference component values. [Table 11-9](#) (page 11-49) 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 11-9 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).

11.3.10 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 \(page 11-1\)"](#)

- [Button Template Attributes \(page 11-49\)](#)

11.3.10.1 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.

**See Also:**

"[Understanding Template Options](#) (page 11-8)" and "[Understanding Theme Subscriptions](#) (page 11-17)"

- [Name](#) (page 11-50)
- [Definition](#) (page 11-50)
- [Substitution Strings](#) (page 11-50)

11.3.10.1.1 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.

11.3.10.1.2 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.

11.3.10.1.3 Substitution Strings

Substitution strings are used within subtemplates to reference component values. [Table 11-10](#) (page 11-51) 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 11-10 Button Template Substitution Strings

Referenced From	Substitution String	Description
Hot Template	#BUTTON_ATTRIBUTES#	Button attributes.
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.

11.3.11 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](#) (page 11-52)
- [Legacy Calendar Template Attributes](#) (page 11-52)

See Also:

"[About Supported Calendars](#) (page 10-2)," "[Creating Calendars](#) (page 10-1)," and "[Using Themes](#) (page 11-1)"

11.3.11.1 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 (#).

11.3.11.2 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](#) (page 11-8)" and "[Understanding Theme Subscriptions](#) (page 11-17)"

Calendar Definition

- [Name](#) (page 11-52)
- [Subscription](#) (page 11-52)
- [Comments](#) (page 11-52)

11.3.11.2.1 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.

11.3.11.2.2 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**.

11.3.11.2.3 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.

11.3.12 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](#) (page 11-53)
- [About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help](#) (page 11-55)

See Also:

["Using Themes](#) (page 11-1)"

11.3.12.1 Label Template Attributes

See Also:

["Understanding Template Options](#) (page 11-8)" and ["Understanding Theme Subscriptions](#) (page 11-17)"

This section describes specific sections of the Edit Label Template page. To learn more about a specific attribute, see field-level Help.

- [Name](#) (page 11-53)
- [Definition](#) (page 11-54)
- [Error Display](#) (page 11-54)
- [Field Container](#) (page 11-55)
- [Substitution Strings](#) (page 11-55)

11.3.12.1.1 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.

11.3.12.1.2 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](#) (page 11-55)."
- **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.

11.3.12.1.3 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.

11.3.12.1.4 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.

11.3.12.1.5 Substitution Strings

Lists substitution string usage for this template. Substitution strings are used within subtemplates to reference component values.

11.3.12.2 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>
```

11.3.13 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](#) (page 11-56)
- [List Template Attributes](#) (page 11-57)

**See Also:**

["Creating Lists](#) (page 16-27)" and ["Using Themes](#) (page 11-1)"

11.3.13.1 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](#) (page 16-27)"

11.3.13.2 List Template Attributes

See Also:

"[Understanding Template Options](#) (page 11-8)" and "[Understanding Theme Subscriptions](#) (page 11-17)"

This section describes some sections of the Edit List Template page. To learn more about a specific attribute, see field-level Help.

- [Name](#) (page 11-57)
- [Before List Entry](#) (page 11-57)
- [Template Definition](#) (page 11-57)
- [Template Definitions for First Entry](#) (page 11-58)
- [Before Sublist Entry](#) (page 11-58)
- [Sublist Entry](#) (page 11-58)
- [After Sub List Entry](#) (page 11-59)
- [After List Entry](#) (page 11-59)
- [Substitution Strings](#) (page 11-59)

11.3.13.2.1 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.

11.3.13.2.2 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.

11.3.13.2.3 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.

11.3.13.2.4 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.

11.3.13.2.5 Before Sublist Entry

Enter HTML that displays before any sublist elements.

11.3.13.2.6 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.

11.3.13.2.7 After Sub List Entry

Enter HTML that displays after displaying sublist elements.

11.3.13.2.8 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.

11.3.13.2.9 Substitution Strings

Lists substitution string usage for this template. Substitution strings are used within subtemplates to reference component values.

11.3.14 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](#) (page 11-59)
- [Required Dialog Page Template Attributes](#) (page 11-67)



See Also:

["Using Themes \(page 11-1\)"](#)

11.3.14.1 Page Template Attributes



See Also:

["Understanding Template Options \(page 11-8\)"](#) and ["Understanding Theme Subscriptions \(page 11-17\)"](#)

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](#) (page 11-60)
- [Definition](#) (page 11-60)

- [JavaScript](#) (page 11-61)
- [Cascading Style Sheet](#) (page 11-61)
- [Subtemplate](#) (page 11-62)
- [Image Based Tab Attributes](#) (page 11-62)
- [Layout](#) (page 11-62)
- [Display Points](#) (page 11-63)
- [Dialog](#) (page 11-63)
- [Error Page Template Control](#) (page 11-63)
- [Substitution Strings](#) (page 11-64)

11.3.14.1.1 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](#) (page 6-49)" and "[Required Dialog Page Template Attributes](#) (page 11-67)"

11.3.14.1.2 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.

11.3.14.1.3 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.

11.3.14.1.4 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.

11.3.14.1.5 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.

11.3.14.1.6 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.

11.3.14.1.7 Layout

Specified 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.

11.3.14.1.8 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.

11.3.14.1.9 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 \(page 11-67\)"](#)

11.3.14.1.10 Error Page Template Control

Use this attribute when a page template is designated as an error template.

11.3.14.1.11 Substitution Strings

Table 11-11 (page 11-64) 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 Stings section of the Edit Page Template page.

Table 11-11 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.
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.

Table 11-11 (Cont.) Page Template Substitution Strings

Referenced From	Substitution String	Description
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 (page 6-22)"
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 (page 6-38)"
Body	#NAVIGATION_BAR#	Navigation bar. See Also: " Subtemplate (page 11-62)" 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.
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.

Table 11-11 (Cont.) Page Template Substitution Strings

Referenced From	Substitution String	Description
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.
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).

Table 11-11 (Cont.) Page Template Substitution Strings

Referenced From	Substitution String	Description
JavaScript	#TRIGGERING_ELEMENT#	Triggering Element (for example, the current DOM element).

11.3.14.2 Required Dialog Page Template Attributes

To create a dialog page template, you must define the Page Template attributes described in [Table 11-12](#) (page 11-67).

Table 11-12 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](#) (page 6-49)" and "[Page Template Attributes](#) (page 11-59)"

11.3.15 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](#) (page 11-68)



See Also:

"[Creating Lists of Values at the Application Level](#) (page 16-24)" and "[Using Themes](#) (page 11-1)"

11.3.15.1 Popup List of Values Template Attributes



See Also:

"[Understanding Theme Subscriptions](#) (page 11-17)"

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](#) (page 11-68)
- [Buttons](#) (page 11-69)
- [Window](#) (page 11-69)
- [Pagination](#) (page 11-69)
- [Result Set](#) (page 11-69)
- [Page Attributes](#) (page 11-69)

11.3.15.1.1 Search Field

Use these attributes to specify how a Search field displays. [Table 11-13](#) (page 11-68) describes available Search Field attributes.

Table 11-13 Search Field Attributes

Attribute	Description
Before Field Text	Defines text to display before the popup list of values search field displays.
After Field Text	Displays this text after displaying the search field, the search button, and the close button.
Filter Width	Displays the <code>HTML INPUT TYPE = TEXT</code> widget using this width.
Filter Max Width	Displays the <code>HTML INPUT TYPE = TEXT</code> widget using this maximum width.
Filter Text Attribute	Displays the <code>HTML INPUT TYPE = TEXT</code> widget using these attributes. This is included within the HTML input tag.

11.3.15.1.2 Buttons

Use these attributes to define the button name and attributes for the Find, Close, Next, and Previous buttons.

11.3.15.1.3 Window

Popup lists of values are executed using JavaScript. Use these attribute to control the values of `scrollbars=`, `resizable=`, `width=`, and `height=`.

11.3.15.1.4 Pagination

Defines how row count results display.

11.3.15.1.5 Result Set

Use these attributes to define text or HTML to display before and after a result set.

11.3.15.1.6 Page Attributes

Use these attributes to define popup pages.

11.3.16 Region Templates



See Also:

["Understanding Template Options \(page 11-8\)"](#) and ["Understanding Theme Subscriptions \(page 11-17\)"](#)

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 \(page 11-70\)](#)



See Also:

["About Regions \(page 12-4\)"](#) and ["Using Themes \(page 11-1\)"](#)

11.3.16.1 Region Template Attributes



See Also:

"[Understanding Template Options](#) (page 11-8)" and "[Understanding Theme Subscriptions](#) (page 11-17)"

This section describes some attributes on the Edit Region Template page. To learn more about a specific attribute, see field-level Help.

- [Name](#) (page 11-70)
- [Definition](#) (page 11-70)
- [Layout](#) (page 11-71)
- [Sub Regions](#) (page 11-71)
- [Substitution Strings](#) (page 11-71)

11.3.16.1.1 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.

11.3.16.1.2 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:

["Understanding Substitution Strings \(page 2-26\)"](#)

11.3.16.1.3 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.

11.3.16.1.4 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.

11.3.16.1.5 Substitution Strings

Lists substitution string used within templates to reference component values.

11.3.17 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](#) (page 11-72)
- [Report Column Template Attributes for Generic Column Templates](#) (page 11-73)
- [Report Column Template Attributes for Named Column Templates](#) (page 11-77)
- [About Using JavaScript in Column Templates](#) (page 11-80)



See Also:

"[Using Themes](#) (page 11-1)," "[Understanding Template Options](#) (page 11-8)," and "[Understanding Theme Subscriptions](#) (page 11-17)"

11.3.17.1 Generic Column Templates and Named Column Templates

Oracle Application Express includes two types of report templates:

- [Generic Column Templates](#) (page 11-72)
- [Named Column Templates](#) (page 11-73)

11.3.17.1.1 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
```

11.3.17.1.2 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>
```

11.3.17.2 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](#) (page 11-73)
- [Before Rows](#) (page 11-74)
- [Column Headings](#) (page 11-74)
- [Before Each Row](#) (page 11-74)
- [Column Templates](#) (page 11-75)
- [After Each Row](#) (page 11-75)
- [After Rows](#) (page 11-76)
- [Row Highlighting](#) (page 11-76)
- [Pagination Subtemplate](#) (page 11-76)
- [Comments](#) (page 11-77)

11.3.17.2.1 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.

11.3.17.2.2 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.

11.3.17.2.3 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.

11.3.17.2.4 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.

11.3.17.2.5 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 11-14](#) (page 11-75).

Table 11-14 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
```

11.3.17.2.6 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>
```


11.3.17.2.7 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.

11.3.17.2.8 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.

11.3.17.2.9 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 11-17](#) (page 11-80) describes these templates.

Table 11-15 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 11-15 (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>

11.3.17.2.10 Comments

Use this attribute to record comments about this component.

11.3.17.3 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](#) (page 11-77)
- [Subscription](#) (page 11-78)
- [Row Templates](#) (page 11-78)
- [Column Headings](#) (page 11-79)
- [Before first and after last row text](#) (page 11-79)
- [Pagination](#) (page 11-79)
- [Comments](#) (page 11-80)

11.3.17.3.1 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.

11.3.17.3.2 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**.

11.3.17.3.3 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 11-16](#) (page 11-78).

Table 11-16 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
```

11.3.17.3.4 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>
```

11.3.17.3.5 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.

11.3.17.3.6 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 11-17](#) (page 11-80) describes these templates.

Table 11-17 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>

11.3.17.3.7 Comments

Use this attribute to record comments about this component.

11.3.17.4 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 \(page 12-4\)"](#)

11.4 Using Custom 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.



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 \(page 11-3\)"](#) and ["Using Theme Roller \(page 11-6\)"](#).

- [About Uploading Cascading Style Sheets \(page 11-81\)](#)
- [Referencing an Uploaded Cascading Style Sheet in the Page Template \(page 11-81\)](#)

11.4.1 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 \(page 16-12\)"](#) and ["Managing Static Application Files \(page 16-12\)"](#).



See Also:

["Referencing an Uploaded Cascading Style Sheet in the Page Template \(page 11-81\)"](#)

11.4.2 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 \(page 11-81\)"](#)
- ["Creating a New Template \(page 11-35\)"](#)
- ["Managing Templates \(page 11-43\)"](#)
- ["Page Templates \(page 11-59\)"](#)
- ["APP_IMAGES \(page 2-32\)"](#)
- ["WORKSPACE_IMAGES \(page 2-46\)"](#)

12

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 POL/SQL.

Topics:

- [About Page Layout in Oracle Application Express](#) (page 12-1)
- [Optimizing a Page for Printing](#) (page 12-2)
- [Creating a Global Page to Display Components on Every Page](#) (page 12-3)
- [About Regions](#) (page 12-4)
- [Managing Regions in Page Designer](#) (page 12-5)
- [Managing Regions in Legacy Component View](#) (page 12-13)
- [About Incorporating Content from Other Web Sites](#) (page 12-30)
- [About Managing Images](#) (page 12-31)
- [Rendering HTML Using Custom PL/SQL](#) (page 12-31)

See Also:

- ["Adding Navigation](#) (page 14-1)"
- ["Managing the Application User Interface](#) (page 6-36)"
- ["Using Themes and Theme Styles](#) (page 11-1)"

12.1 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 \(page 11-1\)"](#)

12.2 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 \(page 12-2\)](#)
- [Using f?p Syntax to Toggle to Printer Friendly Mode \(page 12-3\)](#)

**See Also:**

["Editing Themes \(page 11-10\)"](#) and ["Creating a Theme \(page 11-20\)"](#)

12.2.1 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 \(page 11-10\)"](#)

12.2.2 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 \(page 2-17\)"](#)

12.3 Creating a Global Page to Display Components on Every 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 \(page 12-3\)](#)
- [Navigating to the Global Page \(page 12-4\)](#)

 **See Also:**

["Managing Pages in a Database Application \(page 6-44\)"](#), ["Understanding Conditional Rendering and Processing \(page 2-10\)"](#) and ["Available Conditions \(page C-1\)"](#)

12.3.1 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 \(page 6-36\)"](#)

12.3.2 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).

12.4 About Regions

Each page in an Oracle Application Express application contains one or more regions. A region is an 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 in Page Designer \(page 12-5\)"](#)
- ["Managing Regions in Legacy Component View \(page 12-13\)"](#)
- ["Accessing Page Specific Utilities \(page 6-79\)"](#)

12.5 Managing Regions in Page Designer

Regions function as containers for content. Each page in an Oracle Application Express application contains one or more regions.

- [Creating a Region in Page Designer \(page 12-5\)](#)
- [Managing Regions in Page Designer \(page 12-6\)](#)
- [Copying a Region in Page Designer \(page 12-12\)](#)
- [Deleting Regions in Page Designer \(page 12-13\)](#)

 **See Also:**

["About Regions \(page 12-4\)"](#) and ["Managing Regions in Legacy Component View \(page 12-13\)"](#)

12.5.1 Creating a Region in Page Designer

 **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 \(page 7-15\)"](#)

12.5.2 Managing Regions in Page Designer

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 in Page Designer \(page 12-7\)](#)
- [Controlling Region Positioning \(page 12-8\)](#)
- [Specifying a Region Header and Footer \(page 12-8\)](#)
- [Enabling Users to Customize a Page \(page 12-8\)](#)
- [Utilizing Region Caching in Page Designer \(page 12-9\)](#)
- [Specifying a Static Region ID \(page 12-10\)](#)
- [Adding a Region Image \(page 12-11\)](#)
- [Creating a Region Display Selector \(page 12-11\)](#)

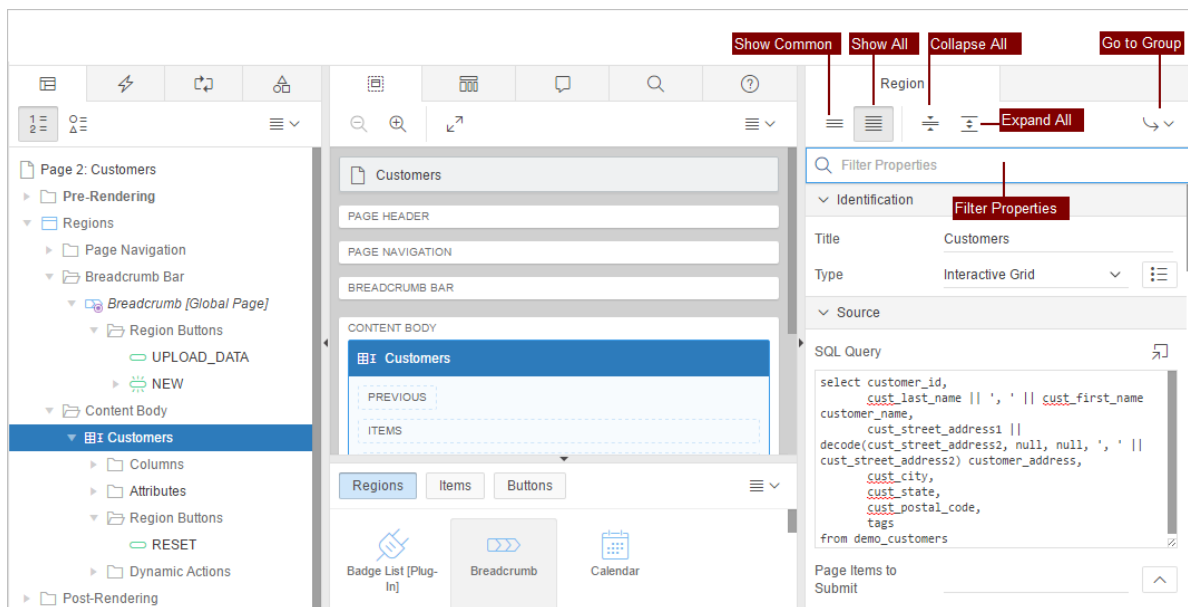
12.5.2.1 Editing Region Attributes in Page Designer

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.



Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

3. View all attributes in the Property Editor. Click the **Show All** icon at the top of the left pane.
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) (page 7-18)"

12.5.2.2 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.

12.5.2.3 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:

```
Fetched #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
```

12.5.2.4 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 region.
3. Locate **Customization**. For Customizable, select one of the following:

- **Not Customizable By End Users**
 - **Customizable and Not Shown By Default**
 - **Customizable and Shown By Default**
4. Click **Save**.



See Also:

"[Creating Custom Themes](#) (page 11-32)"

12.5.2.5 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](#) (page 12-9)
- [Enabling Region Caching in Page Designer](#) (page 12-10)



See Also:

"[Managing Page and Region Caching](#)" in *Oracle Application Express Administration Guide*

12.5.2.5.1 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.



See Also:

"[Server Cache](#) (page 24-23)"

12.5.2.5.2 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 Region attributes.
3. To view all attributes in the Property Editor, click **Show All**.
4. Expand **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**.

12.5.2.6 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 the Region attributes.

3. To view all attributes in the Property Editor, click **Show All**.
4. Locate **Advanced**. Edit the **Static ID** attribute:
 - a. Locate **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**.

12.5.2.7 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.
3. Locate **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.
4. Click **Save**.

12.5.2.8 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. 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.

5. Click **Save**.

12.5.3 Copying a Region in Page Designer

 **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.

12.5.4 Deleting Regions in Page Designer

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**.

12.6 Managing Regions in Legacy Component View

Regions function as containers for content. Each page in an Oracle Application Express application contains one or more regions.

- [Region Types When Running the Create Region Wizard](#) (page 12-13)
- [Creating a Region in Component View](#) (page 12-18)
- [Managing Regions in Component View](#) (page 12-18)
- [Creating a Region Display Selector](#) (page 12-28)
- [Copying a Region in Component View](#) (page 12-28)
- [Deleting Regions in Component View](#) (page 12-29)

 **See Also:**

"[Region Types When Running the Create Region Wizard](#) (page 12-13), "[About Regions](#) (page 12-4)," and "[Managing Regions in Page Designer](#) (page 12-5)"

12.6.1 Region Types When Running the Create Region Wizard

When you create an application, you choose a target user interface. The user interface you select determines the available region type.

Region Types for Desktop Applications

If you are creating a desktop application, the Create Region Wizard supports the following regions.

Table 12-1 Create Region Wizard — Region Types for Desktop Applications

Region Type	Description
Static Content	Functions as containers for items and contains the HTML you provide. Any HTML you type may contain substitution strings.
Multiple Static Content	Use this option to create multiple Static Content regions at once. In the fields provided, specify the Sequence, Title, Display Point, Report Template, and Column for each region.
Report, Interactive Report	Formatted result of a SQL query. You provide a custom SQL SELECT statement or a PL/SQL function returning a SQL Query statement. For assistance in writing the SQL query, click the Query Builder button. Note: jQuery Mobile Smartphone does not support interactive reports. See Also: "Customizing Interactive Reports in a Running Application" (page 8-92)"
Report, Classic Report	Formatted result of a SQL query. You provide a custom SQL SELECT statement or a PL/SQL function returning a SQL Query. For assistance in writing the SQL query, click the Query Builder button. See Also: "Developing Reports" (page 8-1)"
Report, Web Service Result	Creates a report on a Web Service result. See Also: "Implementing Web Services" (page 17-8)"
Form, 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.
Form, Form on a Table or View	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.
Form, Form on a Table	Creates two pages. One page displays a report. Each row provides a link to the second page to enable users to update each record. You can select the table on which to build the report and form. Tip: This option does not support tables having more than 127 columns. Selecting more than 127 columns generates an error.
Form, Form on a SQL Query	Creates a form based on the columns returned by a SQL query such as an <code>EQUIJOIN</code> . For assistance in writing the SQL query, click Query Builder.

Table 12-1 (Cont.) Create Region Wizard — Region Types for Desktop Applications

Region Type	Description
Form, 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 Also: " Creating an Input Form and Report on a Web Service (page 17-17)"
Plug-ins	Plug-ins allow developers to declaratively extend the built-in types available with Application Express. See Also: " Implementing Plug-ins (page 18-3)"
Chart	Create line, bar, or pie charts based on SQL queries. See Also: " Managing Legacy Charts (page 24-32)"
Map Chart	Create declaratively defined Flash maps. See Also: " Creating Maps (page 10-12)"
Tree	Create hierarchical navigational control (or tree) based on a SQL query executed at runtime. See Also: " Creating Trees (page 13-61)"
Calendar	Create a region to contain a calendar with monthly, weekly, daily, and list views. See Also: " Creating Calendars (page 10-1)"
List	Create a list region containing a shared collection of links called list. See Also: " Creating Lists (page 16-27)"
Breadcrumb	Create a region to contain a breadcrumb. Breadcrumbs are a hierarchical list of links. See Also: " Creating Breadcrumbs (page 14-8)"
PL/SQL Dynamic Content	Create a regions based on PL/SQL that enable you to render any HTML or text using the PL/SQL Web Toolkit.
URL	Create a URL based-regions that obtains content by calling a web server using a predefined URL. See Also: " About Incorporating Content from Other Web Sites (page 12-30)"
Region Display Selector	Region Display Selector region enables the display of show hide controls for each region on a page for which region display selection has been enabled. See Also: " Creating a Region Display Selector (page 12-28)"
Help Text	Help Text regions enable you to provide page-level help for your application. See Also: " Creating Help for Your Application (page 10-33)"

Region Types for Mobile Applications

If you are creating a mobile application, the Create Region Wizard supports the following regions.

Table 12-2 Create Region Wizard — Region Types for Mobile Applications

Region Type	Description
Static Content	Functions as containers for items and contains the HTML you provide. Any HTML you type may contain substitution strings.
Multiple Static Content	Use this option to create multiple Static Content regions at once. In the fields provided, specify the Sequence, Title, Display Point, Report Template, and Column for each region.
Report, Classic Report	Formatted result of a SQL query. You provide a custom SQL SELECT statement or a PL/SQL function returning a SQL Query. For assistance in writing the SQL query, click the Query Builder button. See Also: " Developing Reports (page 8-1)" and
Report, Web Service Result	Creates a report on a Web Service result. See Also: " Implementing Web Services (page 17-8)"
Report, List View	List view is optimized to display data and provide easy navigation on Smartphones. Creates a page that contains the formatted result of a SQL query. You provide a custom SQL SELECT statement or a PL/SQL function returning a SQL Query. Note: Desktop applications do not support list view.
Report, Column Toggle Report	Creates a responsive report for mobile applications. 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. Note: Desktop applications do not support this report format.
Reflow Report	Creates a responsive report for mobile applications. When there is not enough space available to display the report horizontally, the report works by collapsing the table columns into a stacked presentation that looks like blocks of label and data pairs for each row. Note: Desktop applications do not support this report format.
Form, 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.

Table 12-2 (Cont.) Create Region Wizard — Region Types for Mobile Applications

Region Type	Description
Form, Form on a Table or View	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.
Form, Form on a Table with List View	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.
Form, Form on a SQL Query	Creates a form based on the columns returned by a SQL query such as an <code>EQUIJOIN</code> . For assistance in writing the SQL query, click Query Builder.
Form, Summary Page	Creates a read-only version of a form. Use this option to create a confirmation page at the end of a wizard.
Form, 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 Also: " Creating a Form on a Web Service (page 17-20)"
Form, Form and Report 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 Also: " Creating an Input Form and Report on a Web Service (page 17-17)"
Plug-ins	Plug-ins allow developers to declaratively extend the built-in types available with Application Express. See Also: " Implementing Plug-ins (page 18-3)"
Chart	Create line, bar, or pie charts based on SQL queries. See Also: " Managing Legacy Charts (page 24-32)"
Calendar	Create a region to contain a calendar with monthly, weekly, daily, and list views. See Also: " Creating Calendars (page 10-1)"
Breadcrumb	Create a region to contain a breadcrumb. Breadcrumbs are a hierarchical list of links. See Also: " Creating Breadcrumbs (page 14-8)"
PL/SQL Dynamic Content	Create a regions based on PL/SQL that enable you to render any HTML or text using the PL/SQL Web Toolkit.

Table 12-2 (Cont.) Create Region Wizard — Region Types for Mobile Applications

Region Type	Description
URL	Create a URL based-regions that obtains content by calling a web server using a predefined URL. See Also: " About Incorporating Content from Other Web Sites (page 12-30)"
Region Display Selector	Region Display Selector region enables the display of show hide controls for each region on a page for which region display selection has been enabled. See Also: " Creating a Region Display Selector (page 12-28)"
Help Text	Help Text regions enable you to provide page-level help for your application. See Also: " Creating Help for Your Application (page 10-33)"

12.6.2 Creating a Region in Component View

You create regions by running the Create Region Wizard.

To create a region:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Regions, click the **Create** icon.
The Create Region Wizard appears.
3. Select a region type. To learn more, see "[Region Types When Running the Create Region Wizard](#) (page 12-13)."
The Create Region Wizard appears.
4. Follow the on-screen instructions.

12.6.3 Managing Regions in Component View

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 in Component View](#) (page 12-19)
- [How Region Attributes Affect Page Layout](#) (page 12-19)
- [Specifying a Region Header and Footer](#) (page 12-21)
- [Enabling Users to Customize a Page](#) (page 12-22)
- [Utilizing Region Caching in Component View](#) (page 12-22)
- [Specifying a Static Region ID](#) (page 12-24)
- [Adding a Region Image](#) (page 12-25)
- [Editing Multiple Regions at Once in Component View](#) (page 12-25)

- [Managing Subregions in Component View](#) (page 12-26)

12.6.3.1 Editing Region Attributes in Component View

To edit region attributes in Component View:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Regions, select the region name.

The Edit Region page (or Region Definition page) appears.

Note:

For further information regarding regions that are plug-in type regions, see "[Importing a Plug-in from the Plug-in Page](#) (page 18-11)."

3. Edit the appropriate attributes.

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

4. Click **Apply Changes**.

See Also:

"[How Region Attributes Affect Page Layout](#) (page 12-19)," "[Controlling Region Positioning](#) (page 12-8)," and "[Specifying a Region Header and Footer](#) (page 12-21)"

12.6.3.2 How Region Attributes Affect Page Layout

This section explains how region attributes control the appearance and layout of a page.

- [Component View Region Attributes Affecting Page Layout](#) (page 12-19)
- [Controlling Region Positioning in Component View](#) (page 12-21)

12.6.3.2.1 Component View Region Attributes Affecting Page Layout

[Table 12-3](#) (page 12-19) describes region attributes that affect the layout of a page.

Table 12-3 Region Attributes Affecting Page Layout

Region, Attribute	Description
User Interface, Sequence	Specifies the display order of the regions within the page.
User Interface, Parent Regions	Defines the parent region to which the current region belongs. By selecting a Parent region, you create a parent/child relationship between two regions. See Also: " Managing Subregions in Component View (page 12-26)"

Table 12-3 (Cont.) Region Attributes Affecting Page Layout

Region, Attribute	Description
User Interface, Display Point	Identifies where the region displays within the page. Regions are rendered in order of sequence number within a Display Point. Click the View icon to see the page layout and select a position. The possible display points for a region are determined by the page-level template (which is a page attribute). If no page-level template is selected, the default page-level template, defined in the Application Definition, is used.
User Interface, Template	Determines the look of the region. To view template attributes, click the template name on the page. See Also: "Creating Custom Themes (page 11-32)" and "Region Templates (page 11-69)"
User Interface, Item Display Position	Determines if page items are displayed above or below the main region content.
Grid Layout, Start New Grid	Determines whether this region displays in a new grid or in the same grid as the previous region.
Grid Layout, Start New Row	Determines whether this region displays on the same row as the previous region or whether it displays on the next row.
Grid Layout, Column	Determines if the region displays in a specific grid column. Select Automatic to automatically determine the next available grid column where the region gets displayed.
Grid Layout, Column Span	Determines how many grid columns should be occupied by the region. Select Automatic to balance the available grid columns within all regions in the same grid row.
Grid Layout, Column CSS Classes	Enter additional CSS classes which are set for the grid column. This value replaces the #CSS_CLASSES# substitution placeholder in the grid layout column template.
Grid Layout, Column Attributes	Enter additional attributes which are set for the grid column. This value replaces the #ATTRIBUTES# substitution placeholder in the grid layout column template.
Attributes, Region CSS Classes	Enter additional CSS classes as space delimited list which are substituted if the region template contains the substitution string #REGION_CSS_CLASSES#.
Attributes, Region	Region templates that contain the substitution string #REGION_ATTRIBUTES# are substituted with this value.
Attributes, Region Display Selector	Use the region display selector to identify the region to display in a region display selector region type. Region display selectors enable you to hide other regions on the page and just show a specific region. Region display selectors only display regions that have been identified using this attribute.
Attributes, Region Image	If populated, the identified image displays in the upper left of the region. There is no control over the table tag used to display this image. See Also: "Adding a Region Image (page 12-25)"
Attributes, Image Tag Attributes	Used with the Region Image attributes. Enter attributes to be included in the image html. For example, use this attribute for height and width. Note: If no ALT is defined in the image tag attributes, Oracle Application Express defaults to rendering the image as a decorative image (for example with an empty ALT tag). See Also: "Adding a Region Image (page 12-25)"

Table 12-3 (Cont.) Region Attributes Affecting Page Layout

Region, Attribute	Description
Header and Footer	Specifies HTML text to be displayed at the top of the region (just before the #BODY# content).
Conditions	Defines conditions and appropriate expressions that determine if the region displays. Conditions can reference session state, the currently logged in user, or environment preferences (such as whether a page is in Print View mode). See Also: " Understanding Conditional Rendering and Processing (page 2-10) " and " Optimizing a Page for Printing (page 12-2) "
Read Only, Read Only Condition Type	Select a condition type from the list that must be met in order for this region to render page items as a read-only. Not selecting a condition causes the page items to render normally. Note: Set the condition type to Never to always render the items of this region as enterable fields. This overwrites any read-only condition defined at the page-level.
Customization	Enables end-user customization. To use this feature, you must include the #CUSTOMIZE# substitution string in the Header, Body, or Footer section of the page template. See Also: " Enabling Users to Customize a Page (page 12-22) "

12.6.3.2.2 Controlling Region Positioning in Component View

When you create a region, you must specify its position (or Display Point) for this region within the page template. In addition to Display Point, you can specify the column in which the region is placed. When you place regions in multiple columns, Oracle Application Express automatically renders the necessary HTML to produce a multiple column layout.

12.6.3.3 Specifying a Region Header and Footer

In addition to the body content of a region (which can be a report, a chart, or HTML with form elements), you can specify additional HTML to be placed above and below a region or in its header and footer.

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
```



See Also:

["Region Types When Running the Create Region Wizard \(page 12-13\)"](#)

12.6.3.4 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 link called Customize 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.



See Also:

["Creating Custom Themes \(page 11-32\)"](#)

To enable end-user customization:

1. Navigate to the page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)".
2. Under Regions, select the region name.
The Edit Region page appears.
3. Scroll down to Customization and select one of the following:
 - **Customizable and Not Shown By Default**
 - **Customizable and Shown By Default**
 - **Not Customizable By End Users**
4. In Customized Option Name, enter the label that represents this region on the page to the user.
5. Include the #CUSTOMIZE# substitution string in the Header, Body, or Footer section of the page template.

12.6.3.5 Utilizing Region Caching in Component View

This section describes the advantages of using caching to improve the performance of pages and regions.



Tip:

Region caching is not available for interactive report regions.

- [About Region Caching](#) (page 12-23)
- [Enabling Region Caching](#) (page 12-23)
- [Managing Cached Regions](#) (page 12-24)



See Also:

"Managing Page and Region Caching" in *Oracle Application Express Administration Guide*

12.6.3.5.1 About Region Caching

Enabling region caching is an effective way 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.



See Also:

"[Server Cache](#) (page 24-23)" and "[Managing Cached Regions](#) (page 12-24)"

12.6.3.5.2 Enabling Region Caching

To enable region caching:

1. Navigate to the page.
See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Regions, select the region name.
The Edit Region page appears.
3. Scroll down to Caching.
4. For Caching, select one of the following:
 - **Not Cached** - Disables this feature.
 - **Cached** - Caches the region independent of the user.
 - **Cached by User** - Caches the region specific to a given user.

5. If you select **Cached** or **Cached by User**, specify the following:
 - a. Timeout Cache After - Identify how long the cached region remains valid.
 - b. Cache Condition Type - Select a condition type from the list. If the condition returns false, the region is rendered dynamically and is not cached. If the condition returns true, the region is cached.
 - c. Expression 1 and Expression 2 - Enter values based on the specific condition type selected.
6. Click **Apply Changes**.

12.6.3.5.3 Managing Cached Regions

You can manage cached regions on the Cached Regions page.

To access the Cached Regions page:

1. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Click the **Utilities** menu and select **Caching**.
3. Under Region Cache, click **Manage**.
The Cached Regions page appears.
4. You can customize the appearance the page using the Search bar at the top of the page.
5. Select the regions you want to purge and click one of the following:
 - **Purge Expired**
 - **Purge Checked**
6. To purge all cached regions, click **Purge All**.



See Also:

"Managing Page and Region Caching" in *Oracle Application Express Administration Guide*

12.6.3.6 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. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".

2. Under Regions, select the region name.
The Edit Region page appears.
3. Scroll down to Attributes.
4. For Static ID, enter value to identify this region. You can reference this value using the substitution string `#REGION_STATIC_ID#`.
5. Click **Apply Changes**.

12.6.3.7 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. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Regions, select the region name.
The Edit Region page appears.
3. Scroll down to Attributes.
4. Under Attributes, specify the following:
 - Region Image - Specify an image to display in the upper left of the region.
 - Image Tag Attributes - Enter attributes to be included in the image html. Use these attributes to specify height and width.
5. Click **Apply Changes**.

12.6.3.8 Editing Multiple Regions at Once in Component View

This section describes ways to edit multiple regions at once.

- [Editing Multiple Attributes Across All Regions](#) (page 12-25)
- [Accessing Region Utilities](#) (page 12-26)

12.6.3.8.1 Editing Multiple Attributes Across All Regions

Using the Regions page you can update multiple region attributes across all regions on a page, including sequence (the order), column, region name, and display point.

See Also:

To learn more about region attributes, go to the Edit Region page. See "[Managing Regions in Component View](#) (page 12-18)".

To edit multiple attributes across all regions:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".

2. Under Regions, click the **Edit All** icon. The Edit All icon resembles a small grid with a pencil on top of it.
Regions page appears.
3. Update the appropriate attributes.
4. Click **Apply Changes**.

12.6.3.8.2 Accessing Region Utilities

Use the Utilities page to access Grid Edit pages and reports across all pages within a selected application.

To access the Region Utilities page:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Regions, click the **Edit All** icon. The Edit All icon resembles a small grid with a pencil on top of it.
Regions page appears.
3. Click **Utilities**.
4. Select a utility and follow the on-screen instructions.



See Also:

["Managing Regions in Component View](#) (page 12-18)"

12.6.3.9 Managing Subregions in Component View

Subregions enable you to create a parent/child relationship between two regions. A subregion displays inside of a parent region. You create a subregion by selecting a parent region either when you initially create the region, or when editing region attributes.

- [About Using Subregions](#) (page 12-26)
- [Creating a Subregion in Component View](#) (page 12-27)
- [Creating a Subregion by Editing Region Attributes](#) (page 12-27)

12.6.3.9.1 About Using Subregions

You can use this subregions in a variety of ways. One use case would be to create subregions to group information. For example, suppose you have a region with a border. You can use this feature to display two classic reports (or subregions) within it. You can also use subregions to display regions as tabs or as an accordion. You control how parent and subregions display by editing the Sub Region attributes of the region template.



See Also:

["Region Templates \(page 11-69\)"](#)

12.6.3.9.2 Creating a Subregion in Component View

To create a subregion:

1. Navigate to the appropriate page. See ["Viewing a Page in Legacy Component View \(page 24-10\)"](#).
2. Under Regions, click the **Create** icon.
3. Select a region type. See [Region Types When Running the Create Region Wizard \(page 12-13\)](#)

The Create Region Wizard appears.

4. For Display Attributes:
 - a. Title - Enter a title for the region. This title displays if the region template you choose displays the region title.
 - b. Region Template - Enter a title for the region. This title displays if the region template you choose displays the region title.
 - c. Parent Region - Select the parent region to which the new region belongs.
 - d. Display Point - Identify a display point for this region. There are two types of display points: Page template positions and Page body positions.

Page template positions provide for exact placement of a region within a page template. Page template positions are controlled by page template substitution strings (#REGION_POSITION_01#..#REGION_POSITION_08#). Page body positions are displayed where the #BODY# substitution string in the page template indicates.

- e. Click **Next**.
5. Follow the on-screen instructions.

12.6.3.9.3 Creating a Subregion by Editing Region Attributes

To creating a subregion by editing region attributes:

1. Navigate to the appropriate page. See ["Viewing a Page in Legacy Component View \(page 24-10\)"](#).
2. Under Regions, select the region name.
The Edit Region page appears.
3. Scroll down to User Interface.
4. For Parent Region, select the parent region to which the region belongs.
5. If needed, edit additional attributes.

To learn more about an attribute, see field-level Help. See ["Viewing Field-Level Help \(page 1-27\)"](#).

6. Click **Apply Changes**.



See Also:

"[How Region Attributes Affect Page Layout](#) (page 12-19)," "[Controlling Region Positioning](#) (page 12-8)," and "[Specifying a Region Header and Footer](#) (page 12-21)"

12.6.4 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, edit the Region Display Selector attribute:
 - a. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
 - b. Under Regions, select the region name.
The Edit Region page appears.
 - c. Under Attributes, change Region Display Selector to **Yes**.
 - d. Click **Apply Changes**.
3. Create a region of type Region Display Selector:
 - a. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
 - b. Under Regions, click the **Create** icon.
The Create Region Wizard appears.
 - c. Select the region type **Region Display Selector**.
 - d. Click **Create**.

12.6.5 Copying a Region in Component View



Note:

You cannot copy a Tree region since this type of region encompasses more than one region.

To copy a region:

1. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Regions, click the **Copy** icon. The Copy icon resembles two small overlapping pages.

The Copy Region Wizard appears.

3. Select the region you want to copy.
4. For To Page:
 - a. From To Page, select the page to which you want to copy the region.
 - b. Select the button or region items (if applicable) to be copied by selecting **Yes** or **No**.
 - c. Click **Next**.
5. Click **Copy Region**.

12.6.6 Deleting Regions in Component View

This section describes how to delete regions.

- [Deleting a Region](#) (page 12-29)
- [Deleting Multiple Regions at Once](#) (page 12-29)

12.6.6.1 Deleting a Region

To delete a region:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Under Regions, select the region name.
The Edit Region page appears.
3. Click **Delete**.

12.6.6.2 Deleting Multiple Regions at Once

To delete multiple regions at once:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)".
2. Click the **Edit All** icon. The Edit All icon resembles a small grid with a pencil on top of it.
Regions page appears.
3. Click the **Delete Multiple Regions** tab.
4. Specify the following:
 - a. Cascade to Delete Button - Select **Yes** to delete corresponding region buttons.
 - b. Cascade to Delete Items - Select **Yes** to delete corresponding region items.
 - c. Delete associated list and list Entries - Select **Yes** to delete corresponding lists. This option only displays when the region to be deleted is the only region which references the list.
 - d. Cascade to Delete Dynamic Actions - Select **Yes** to delete corresponding region items.
5. Select the regions to be deleted.

6. Click **Apply Changes**.

12.7 About Incorporating Content from Other Web Sites

Typically, pages in an application are based on data stored in an Oracle database. To incorporate content from other servers, you can create a region based on a URL to display content. 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](#) (page 6-16)."

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 in Page Designer](#) (page 12-7)" and "[Editing Region Attributes in Component View](#) (page 12-19)"

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*

12.8 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](#) (page 16-12)" and "[Managing Static Workspace Files](#) (page 16-17)"

12.9 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

13

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.

Topics:

- [Understanding Page-Level Items](#) (page 13-1)
- [Managing Page-Level Items in Page Designer](#) (page 13-8)
- [Managing Page-Level Items in Legacy Component View](#) (page 13-20)
- [Managing Dynamic Actions](#) (page 13-36)
- [Creating Buttons](#) (page 13-51)
- [Creating Trees](#) (page 13-61)
- [About Incorporating JavaScript into an Application](#) (page 13-71)

See Also:

- ["Controlling Page Layout](#) (page 12-1)"
- ["Adding Navigation](#) (page 14-1)"
- ["Manually Refreshing Oracle Application Express Components](#) (page 18-19)"

13.1 Understanding Page-Level Items

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. 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.

- [About the Differences Between Page Items and Application Items](#) (page 13-2)
- [About Item Naming Conventions](#) (page 13-2)
- [Referencing Item Values](#) (page 13-2)
- [About Referencing Items Using JavaScript](#) (page 13-4)
- [Working with Multiple Select List Item](#) (page 13-4)



See Also:

"[Managing Page-Level Items in Page Designer](#) (page 13-8)" and "[Managing Page-Level Items in Legacy Component View](#) (page 13-20)"

13.1.1 About the Differences Between Page Items and Application Items

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. **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](#) (page 16-20)"

13.1.2 About 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](#) (page 2-24)."
- 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`).

13.1.3 Referencing Item Values

You can reference item values stored in session state in regions, computations, processes, validation, and branches. [Table 13-1](#) (page 13-3) describes the supported syntax for referencing item values.

Table 13-1 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 <code>v</code> 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 <code>NV</code> 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 \(page 2-22\)"](#)
- ["Understanding Cross-Site Scripting Protection \(page 19-21\)"](#)
- ["Managing Session State Values \(page 2-20\)"](#)

13.1.4 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*

13.1.5 Working with Multiple Select List Item

This section describes how to handle values returned from multiple select list item.

- [About Handling Values Returned from a Multiple Select List Item \(page 13-5\)](#)
- [Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values \(page 13-5\)](#)
- [Creating a Shuttle Item \(page 13-6\)](#)
- [About Handling Values Returned from a Multiple Select List Item \(page 13-5\)](#)
- [Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values \(page 13-5\)](#)
- [Creating a Shuttle Item \(page 13-6\)](#)

13.1.5.1 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

13.1.5.2 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*

13.1.5.3 Creating a Shuttle Item

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.

- [Creating a Report and From on a Table](#) (page 13-6)
- [Creating a Shuttle Item on the Form Page in Page Designer](#) (page 13-7)
- [Creating a Shuttle Item on the Form Page in Component View](#) (page 13-7)

13.1.5.3.1 Creating a Report and From on a Table

The procedures in this section assumes you have already created an application.

To create report and form on a table:

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 **Desktop**.
 - b. Select a page type - Select **Form**.
 - c. Click **Next**.
5. Select **Report with Form on Table** and click **Next**.
6. On Report Page Attributes:
 - a. Edit the required attributes.
To learn more about an option, see field-level Help.
 - b. Click **Next**.
7. On Navigation Menu, select a Navigation Preference and click **Next**.
8. On Data 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 Column(s) to be shown in Report(Value Required) - Select one or more columns to be included in the report.
 - d. Click **Next**.
9. On Form Page:

- a. Primary Key - Choose **Managed by Database (ROWID)** to have the form use the ROWID pseudo column to identify rows to update and delete. Choose **Select Primary Key Column(s)** to use the source table's primary key column(s).
- b. Select Column(s) to be included in Form(Value Required) - Select one or more columns to be included in the form.
- c. Click **Create**.

13.1.5.3.2 Creating a Shuttle Item on the Form Page in Page Designer

To create a shuttle on the form page:

1. View the form page in Page Designer.
2. In Page Rendering, right-click region containing the form and select **Create Page Item**.

Next, enter attributes in Property Editor - Page Item.

3. Under Identification:
 - a. Name - Enter the name of this item (for example, `PX_EMP_LIST`).

 **Tip:**

In this example, `PX` in the item name (for example `P2_EMP_LIST`) indicates the page on which the item resides.

- b. Type - Select **Shuttle**.
4. Under List of Values:
 - a. Type - Select **SQL Query**.
 - b. SQL Statement - Enter the SQL query definition to populate this list of values. For example:

```
SELECT ename, empno FROM emp ORDER BY 1
```

5. Under Source:
 - Type - Select **SQL Query (return colon separated value)**
 - 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.

13.1.5.3.3 Creating a Shuttle Item on the Form Page in Component View

To create a shuttle on the form page in Component View:

1. Go to the form page and view it in Component View. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."

2. Under Items, click **Create**.
3. For Item Type, select Shuttle and click **Next**.
4. For Item Name, enter `PX_EMP_LIST` and click **Next**.

 **Tip:**

In this example, `PX` in the item name (for example `P2_EMP_LIST`) indicates the page on which the item resides.

5. For Item Attributes, accept the defaults and click **Next**.
6. For Settings, accept the defaults and click **Next**.
7. For List of Values:
 - List of Values Query - Enter:

```
SELECT ename, empno FROM emp ORDER BY 1
```
 - Click **Next**.
8. For List of Values:
 - Source Type - Select **SQL Query (return colon separated value)**
 - Item Source Value - Enter:

```
SELECT empno FROM emp WHERE deptno = :P8_DEPTNO ORDER BY ename
```
 - Click **Create Item**.
9. Click **Run** to view the page. Your shuttle item should look similar to the following illustration.

13.2 Managing Page-Level Items in Page Designer

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. This section describes how to create page-level items in Page Designer.

- [Viewing Page-Level Items in Page Designer](#) (page 13-9)
- [Creating Page-Level Items](#) (page 13-9)
- [Editing Page-Level Items in Page Designer](#) (page 13-14)

 **See Also:**

"[Understanding Page-Level Items](#) (page 13-1)" and "[Managing Page-Level Items in Legacy Component View](#) (page 13-20)"

13.2.1 Viewing Page-Level Items 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.

3. 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.

4. Click **Save**.
5. Click **Save and Run** to view the page.

13.2.2 Creating Page-Level Items

This section describes how to create page-level items by running the Create Item Wizard.

- [Creating a Page-Level Item from the Rendering Tab](#) (page 13-10)
- [Adding a Page-Level Item from the Gallery](#) (page 13-10)
- [Creating a Static List of Values](#) (page 13-11)
- [Creating a Cascading List of Values](#) (page 13-13)

See Also:

["About the Differences Between Page Items and Application Items \(page 13-2\)"](#)

13.2.2.1 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**.
4. To view all attributes in the Property Editor, click **Show All**.
5. To search for a group or an attribute, enter a keyword in the Filter Properties field.
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 center pane.

7. Click **Save**.

 **See Also:**

"[Viewing a Page in Page Designer](#) (page 7-22)" and "[About Regions](#) (page 12-4)"

13.2.2.2 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 \(page 7-22\)"](#) and ["About Regions \(page 12-4\)"](#)

13.2.2.3 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.

The Property Editor displays the Page Item attributes.
3. Edit the appropriate attributes in the Property Editor.

Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.

 **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 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**.
5. Under **List of Values**:
- a. **Name** - Select **Static Values**.
 - b. **Static Values** - Enter the text for this static list of values using the following syntax:

```
STATIC[2]:Display Value[;Return Value],Display Value[;Return Value]
```

 Where:
 - The first keyword may be `STATIC` or `STATIC2`.
 If the text starts with `STATIC:`, the list values displays alphabetically. Alternatively, if the text starts with `STATIC2:`, the list values displays in the same order they are specified in the text.
 - A semicolon separates the display value from the return value in each entry.
 - `Return Value` is optional. If a `Return Value` is not included, the return value equals the display value.

To see examples, click the **Help** tab in the central pane.
6. Click **Save**.
7. Click **Save and Run** to view the page.



See Also:

"[Creating Lists of Values at the Application Level](#) (page 16-24)" and "[About Item Naming Conventions](#) (page 13-2)"

- [Example 1: Four Values Displayed in Alphabetical Order](#) (page 13-12)
- [Example 2: Ten Values Displayed in the Order Listed](#) (page 13-12)
- [Example 3: A List of Values with Having Both a Return and Display Value](#) (page 13-13)

13.2.2.3.1 Example 1: Four Values Displayed in Alphabetical Order

In this example, the list of values has four values (Cow, Dog, Cat, and Lion) that display in alphabetical order. The return value of each entry equals the display value.

```
STATIC:Cow,Dog,Cat,Lion
```

13.2.2.3.2 Example 2: Ten Values Displayed in the Order Listed

In this example, the list of values has ten values that display in the order listed in the definition. The return value of each entry equals the display value.

```
STATIC2:10,15,20,25,50,100,200,500,1000,10000
```

13.2.2.3.3 Example 3: A List of Values with Having Both a Return and Display Value

In this example, the list of values has two values: `Yes` and `No` (the display value `Yes` and its return value `Y`, and the display value `No` and its return value `N`).

```
STATIC:Yes;Y,No;N
```

13.2.2.4 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 and name it `P1_DEPTNO`:
 - a. Create a select list item.
 - b. For Identification, Type, select **Select List**.
 - c. For Identification:
 - Name - Enter:
`P1_DEPTNO`
 - Type - Select **Select List**.
 - d. For Label:
 - Label - Enter:
`Department`
 - e. For List of Values:
 - Type - Select **SQL Query**.
 - For SQL Query, enter:

```
SELECT dname as d,  
       deptno as r  
FROM dept  
ORDER BY dname
```
 - Display Null Value - Select **Yes**.
 - Null Display Value - Enter - Select:
`- Select Department -`
 - f. Accept the remaining defaults.
 - g. Click **Save**.
2. Create the second item named `P1_EMPNO`:

- a. Create a select list item.
- b. For Identification, Type, select **Select List**.
- c. For Identification:
 - Name - Enter:
`P1_EMPNO`
 - Type - Select **Select List**.
- d. For Label:
 - Label - Enter:
`Employee`
- e. For List of Values:
 - Type - Select **SQL Query**.
 - For SQL Query, enter:

```
SELECT ename as d,  
       empno as r  
FROM emp  
WHERE deptno = :P1_DEPTNO  
ORDER BY ename
```
 - Display Null Value - Select **Yes**.
 - Null Display Value - Enter - Select:
`- Select Employee -`
- f. Accept the remaining defaults.
- g. Click **Save**.

Two select lists appear. Making a selection in the Department select list, determines which individuals display in the Employees select list.

13.2.3 Editing Page-Level Items in Page Designer

This section describes how to edit page-level items in Page Designer.

- [Editing Page Item Attributes in the Property Editor](#) (page 13-14)
- [Defining Default Values for Page Items](#) (page 13-15)
- [Configuring Page Item Security](#) (page 13-16)
- [Creating a Quick Pick Selection](#) (page 13-17)
- [Displaying Conditional Page Items](#) (page 13-18)
- [Displaying Read Only Page Items](#) (page 13-18)
- [Configuring Item Attributes to Warn Users of Unsaved Changes](#) (page 13-19)
- [Viewing Item Utilities](#) (page 13-20)

13.2.3.1 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 changes to display Button attributes.
Attributes are organized into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
To search for a group or an attribute, enter a keyword in the Filter Properties field.
3. 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](#) (page 1-27)."

4. Click **Save**.

13.2.3.2 Defining Default Values for Page Items

You define default values for an item using the attributes under Default on the Edit Page Item page. 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.
To search for a group or an attribute, enter a keyword in the Filter Properties field.
3. Locate **Default**.
4. Select a **Type**:
 - **Static Value**
 - **PL/SQL Expression**
 - **PL/SQL Function Body**

5. Click **Save**.

13.2.3.3 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.

3. Locate and expand **Security**. Configure the following Security attributes.

Table 13-2 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 (page 19-59)"
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 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: " Understanding Session State Protection (page 19-26)"

Table 13-2 (Cont.) Property Editor - Page Item, Security

Attribute	Description
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 (page 19-21)"
Restricted Characters	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. Available options include: <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 See Also: " Understanding Cross-Site Scripting Protection (page 19-21)"

13.2.3.4 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.

3. 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.
4. Click **Save**.

13.2.3.5 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.

3. Locate **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.

4. Click **Save**.

13.2.3.6 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.
3. 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.

4. Click **Save**.

13.2.3.7 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.
3. Locate and expand **Advanced**.
4. 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** - Changes to the item are ignored by the unsaved changes check. Select this option items not based on a database column (for example, a search item).
5. 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 \(page 7-36\)"](#) and *Oracle Application Express API Reference*

13.2.3.8 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**

13.3 Managing Page-Level Items in Legacy Component View

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. This section describes how to create page-level items in Component View.

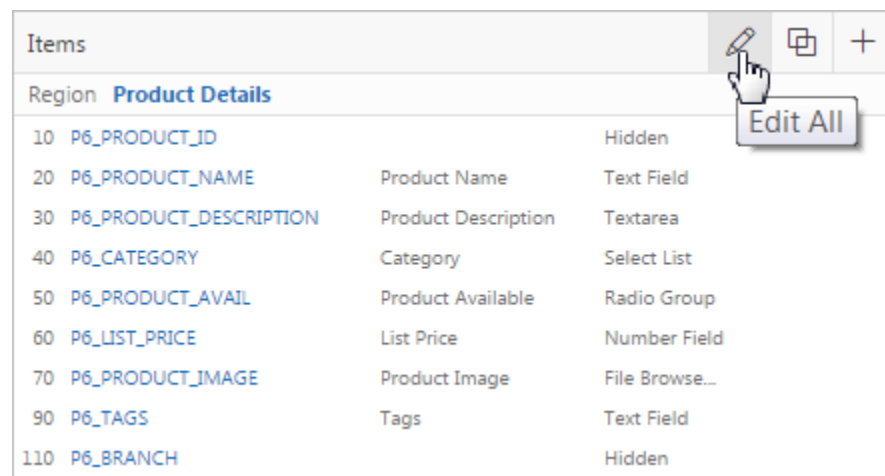
- [Viewing Page-Level Items in the Component View](#) (page 13-21)
- [Creating Page-Level Items in Component View](#) (page 13-22)
- [Editing Page-Level Items in Component View](#) (page 13-28)

 **See Also:**

"[Understanding Page-Level Items](#) (page 13-1)" and "[Managing Page-Level Items in Page Designer](#) (page 13-8)"

13.3.1 Viewing Page-Level Items in the Component View

This section describes how to create and edit page-level items.



Items			
Region	Product Details		
10	P6_PRODUCT_ID		Hidden
20	P6_PRODUCT_NAME	Product Name	Text Field
30	P6_PRODUCT_DESCRIPTION	Product Description	Textarea
40	P6_CATEGORY	Category	Select List
50	P6_PRODUCT_AVAIL	Product Available	Radio Group
60	P6_LIST_PRICE	List Price	Number Field
70	P6_PRODUCT_IMAGE	Product Image	File Browse...
90	P6_TAGS	Tags	Text Field
110	P6_BRANCH		Hidden

In Component View, items display under the section Items. To edit, copy, and create items, click the following icons next to the section title:

- **Edit All** - The Edit All icon resembles a small grid with a pencil on top of it. Use this icon to edit all items at once. See "[Editing Multiple Items Simultaneously](#) (page 13-33)."
- **Copy** - The Copy icon resembles two small overlapping pages. Use this icon to make a copy of an existing item.
- **Create** - The Create icon resembles a plus (+) sign overlapping a small page. Click this icon to create an item.

Items are organized by region. To edit an item, click the item name.

 **See Also:**

"[Editing Page Item Attributes in Component View](#) (page 13-28)"

13.3.2 Creating Page-Level Items in Component View

This section describes how to create page-level items by running the Create Item Wizard.

- [Creating a Page-Level Item in Component View](#) (page 13-22)
- [Creating a Static List of Values](#) (page 13-23)
- [Creating a Cascading List of Values](#) (page 13-25)
- [Creating Multiple Items Using a Tabular Form](#) (page 13-27)



See Also:

["About the Differences Between Page Items and Application Items \(page 13-2\)"](#) and ["About Item Types \(page A-1\)"](#)

13.3.2.1 Creating a Page-Level Item in Component View

To create a page-level item:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. If necessary, create a region. See "[About Regions](#) (page 12-4)."
3. Under Items, click the **Create** icon.
The Create Item Wizard appears.
4. For Create Item:
 - a. Item Type - Select an item type. See "[About Item Types](#) (page A-1)."
When creating a Popup LOV, Color Picker, or Text Field with Calculator Popup, select **Popup LOV** and then the specific popup list of values type.
 - b. Show Unsupported - Selecting **No** only displays items types designed for the user interface of the current page. Selecting **Yes** displays any additional item types which are designed for other user interfaces.
 - c. Click **Next**.
Select an item type. See "[About Item Types](#) (page A-1)."
5. For Display Position and Name:
 - a. Item Name - Enter an item name. This name provides a handle for retrieving the value of the item. Item names longer than 30 characters cannot be referenced using bind variable syntax. See "[About Item Naming Conventions](#) (page 13-2)."
 - b. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - c. Region - Determines the region in which the item is placed.
 - d. Click **Next**.

6. Follow the on-screen instructions.
7. To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

13.3.2.2 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.



See Also:

["Creating Lists of Values at the Application Level](#) (page 16-24)"

To create a static list of values:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. If necessary, create a region. See "[About Regions](#) (page 12-4)."
3. Under Items, click the **Create** icon.
The Create Item Wizard appears.
4. For Create Item:
 - a. Item Type - Select **Select List**.
 - b. Show Unsupported - Selecting **No** only displays items types designed for the user interface of the current page. Selecting **Yes** displays any additional item types which are designed for other user interfaces.
 - c. Click **Next**.
Select an item type. See "[About Item Types](#) (page A-1)."
5. For Display Position and Name:
 - a. Item Name - Enter an item name. This name provides a handle for retrieving the value of the item. Item names longer than 30 characters cannot be referenced using bind variable syntax. See "[About Item Naming Conventions](#) (page 13-2)."
 - b. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - c. Region - Determines the region in which the item is placed.
 - d. Click **Next**.
6. For Item Attributes:
 - a. Label - Enter an item label. The label is the text that appears next to the item when it displays. If you do not want a label, leave the label attribute blank.
 - b. Height - Enter the number of rows high for the item display. For most items this should be 1.
 - c. Template - Optionally, select a field template to control the display of the item.

- d. Click **Next**.
7. For Settings:
- a. Value Required - If set to **Yes** and the page item is visible, Application Express automatically performs a NOT NULL validation when the page is submitted.

 **Tip:**

By defining a message called `APEX.PAGE_ITEM_IS_REQUIRED` in Shared Components/Text Messages, the predefined error text can be replaced by an application specific error text. The display location of the message is defined by the application-level setting Default error display location.

- b. Page Action on Selection - Specify whether multiple values or only a single value can be selected from the list of values. Multiple values are returned as a comma-separated list.
- c. Allow Multi Selection - Specify whether multiple values or only a single value can be selected from the list of values. Multiple values are returned as a comma-separated list.
- d. Click **Next**.

8. For List of Values:

- a. Named LOV - Select **Select Named LOV**.
- b. List of Values Query - Enter a query using the following syntax:

```
STATIC[2]:Display Value[;Return Value],Display Value[;Return Value]
```

Where:

- The first keyword may be `STATIC` or `STATIC2`.
`STATIC` results in the values being sorted alphabetically by display value.
`STATIC2` results in the values being displayed in the order they are entered in the list.
- A semicolon separates the display value from the return value in each entry.
- `Return Value` is optional. If a `Return Value` is not included, the return value equals the display value.

- c. Edit the remaining attributes and click **Next**.

9. Edit the remaining attributes as needed and click **Create Item**.

The examples that follow demonstrate syntax for three different static LOVs.

- [Example 1: Four Values Displayed in Alphabetical Order](#) (page 13-25)
- [Example 2: Ten Values Displayed in the Order Listed](#) (page 13-25)
- [Example 3: A List of Values with Having Both a Return and Display Value](#) (page 13-25)

13.3.2.2.1 Example 1: Four Values Displayed in Alphabetical Order

In this example, the list of values has four values (Cow, Dog, Cat, and Lion) that display in alphabetical order. The return value of each entry equals the display value.

```
STATIC:Cow,Dog,Cat,Lion
```

13.3.2.2.2 Example 2: Ten Values Displayed in the Order Listed

In this example, the list of values has ten values that display in the order listed in the definition. The return value of each entry equals the display value.

```
STATIC2:10,15,20,25,50,100,200,500,1000,10000
```

13.3.2.2.3 Example 3: A List of Values with Having Both a Return and Display Value

In this example, the list of values has two values: Yes and No (the display value Yes and its return value Y, and the display value No and its return value N).

```
STATIC:Yes;Y,No;N
```

13.3.2.3 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. .

Tip:

To learn more about item types, see "[About Item Types \(page A-1\)](#)".

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 sections 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 and name it P1_DEPTNO:
 - a. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)".
 - b. If necessary, create a region. See "[About Regions \(page 12-4\)](#)".
 - c. Under Items, click the **Create** icon.
The Create Item Wizard appears.
 - d. For Item Type, select **Select List**.
 - e. For Display Position and Name:
 - Item Name - Enter:


```
P1_DEPTNO
```

- Accept the remaining defaults and click **Next**.
- f. For Item Attributes:
 - Label - Enter `Department`
 - Accept the remaining defaults and click **Next**.
 - g. For Settings, accept the defaults and click **Next**.
 - h. For List of Values:
 - Display Null Value - Accept the default, **Yes**.
 - Null Display Value - Enter - Select Department -
- Select Department -
 - In List of values definition, enter:


```
SELECT dname as d,
       deptno as r
FROM dept
ORDER BY dname
```
 - Click **Next**.
 - i. Accept the remaining defaults.
 - j. Click **Create Item**.
2. Create the second item named `P1_EMPNO`:
 - a. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
 - b. If necessary, create a region. See "[About Regions](#) (page 12-4)."
 - c. Under Items, click the **Create** icon.
The Create Item Wizard appears.
 - d. For Item Type, select **Select List**.
 - e. For Display Position and Name:
 - Item Name - Enter:


```
P1_EMPNO
```
 - Accept the remaining defaults and click **Next**.
 - f. For Item Attributes:
 - Label - Enter `Employee`.
 - Accept the remaining defaults and click **Next**.
 - g. For Settings, accept the defaults and click **Next**.
 - h. For List of Values:
 - Display Null Value - Accept the default, **Yes**.
 - Null Display Value - Enter - Select Department -
- Select Employee -
 - From Cascading LOV Parent Item(s) - Select **P1_DEPTNO**.

- In List of values definition, enter:

```
SELECT ename as d,
       empno as r
FROM emp
WHERE deptno = :P1_DEPTNO
ORDER BY ename
```

- Click **Next**.
 - i. Accept the remaining defaults.
 - j. Click **Create Item**.
3. Click Run to view a rendered version of the page.

Two select lists appear. Making a selection in the Department select list, determines which individuals display in the Employees select list.

The screenshot shows a web page titled "Cascading LOV Example". It contains two dropdown menus. The first dropdown is labeled "Department" and has "RESEARCH" selected. The second dropdown is labeled "Employee" and has a dropdown menu open showing the following options: "- Select Employee -", "- Select Employee -", "ADAMS", "FORD", "JONES", "SCOTT", and "SMITH".

See Also:

["Editing Page-Level Items in Component View \(page 13-28\)"](#)

13.3.2.4 Creating Multiple Items Using a Tabular Form

To create multiple items simultaneously:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)."
2. If necessary, create a region. See "[About Regions \(page 12-4\)](#)."
3. Access the Create Item Wizard. Under Items, click the **Create** icon.
The Create Item Wizard appears.
4. Click the **Create multiple items using tabular form** link at the bottom of the page.

5. On the Create Multiple Items page, specify the following:
 - a. Create Item(s) in Region - Select the region to contain the items.
 - b. Item Template - Select an item template.
 - c. For each item, enter the Sequence, Name, Label, Type and specify whether the item should be cached.
6. Click **Create Multiple Items**.

13.3.3 Editing Page-Level Items in Component View

You can edit page-level items by editing page item attributes or using the Edit All or Reorder Items icons.

- [Editing Page Item Attributes in Component View](#) (page 13-28)
- [Defining Default Values](#) (page 13-29)
- [Configuring Page Item Security](#) (page 13-29)
- [Creating a Quick Pick Selection](#) (page 13-31)
- [Displaying Conditional Page Items](#) (page 13-32)
- [Displaying Read Only Page Items](#) (page 13-32)
- [Editing Multiple Items Simultaneously](#) (page 13-33)



See Also:

["About Item Types](#) (page A-1)"

13.3.3.1 Editing Page Item Attributes in Component View

Once you create a page item, you can edit it on the Edit Page Item page.

To edit page item attributes:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Items, select the item name.
The Edit Page Item page appears.
3. To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."
4. Click **Apply Changes**.



See Also:

["Understanding Cross-Site Scripting Protection](#) (page 19-21)"

13.3.3.2 Defining Default Values

You define default values for an item using the attributes under Default on the Edit Page Item page. 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. Create the item. See "[Creating Page-Level Items in Component View](#) (page 13-22)."
2. View the item attributes. Under Items, select the item name.
The Edit Page Item page appears.
3. Scroll down to Default.
4. From Default Value Type, select one of the following:
 - **Static Text with Session State Substitutions**
 - **PL/SQL Function Body**
 - **PL/SQL Expression**
5. In Default Value, enter a value or code based on your selection.
6. Click **Apply Changes**.

13.3.3.3 Configuring Page Item Security

You can configure security for an item in Security region on the Edit Page Item page.

To configure page item security:

1. Create the item. See "[Creating Page-Level Items in Component View](#) (page 13-22)."
2. View the item attributes. Under Items, select the item name.
The Edit Page Item page appears.
3. Scroll down to Security.
4. Configure the attributes described in [Table 13-3](#) (page 13-29).

Table 13-3 Edit 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 (page 19-59)"

Table 13-3 (Cont.) Edit Page Item, Security

Attribute	Description
Session State Protection	<p>Select a Session State Protection level. Options include:</p> <ul style="list-style-type: none"> • Unrestricted - The item may be set by passing the item in a URL or in a form. No checksum is required in the URL. • Restricted: May not be set from browser - The item may not be altered using the URL or POSTDATA. Select this option to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute only applies to items that cannot be used as data entry items and is always observed, even if Session State Protection is disabled. <p>Use this attribute for application items or for page items having the following Display As types: Display Only (Save State=No), Text Field (Disabled, does not save state), and Stop and Start Grid Layout (Displays label only).</p> <p>If Session State Protection is enabled, these additional attributes may be applied to items:</p> <ul style="list-style-type: none"> • Checksum Required: Application Level - The item may be set by passing the item in a URL if a checksum is also provided that is specific to the workspace and application. • Checksum Required: User Level - The item may be set by passing the item in a URL if a checksum is also provided that is specific to the workspace, application, and user. • Checksum Required: Session Level - The item may be set by passing the item in a URL if a checksum is also provided that is specific to the current session. <p>See Also: "Understanding Session State Protection (page 19-26)"</p>
Store value encrypted in session state	<p>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.</p> <p>Values up to 4000 bytes in length can be encrypted. Attempts to encrypt values longer than 4000 bytes produce an error message.</p>
Escape Special Characters	<p>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.</p> <p>See Also: "Understanding Cross-Site Scripting Protection (page 19-21)"</p>

Table 13-3 (Cont.) Edit Page Item, Security

Attribute	Description
Restricted Characters	<p>Use this attribute to restrict the characters that can be saved in session state. Oracle Application Express displays an error message if a user tries to save data which does not conform to the selected character restriction. Available options include:</p> <ul style="list-style-type: none"> • 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 &<>"' ; * = % and -- and new line Do not allow &, <, >, ", /, ;, ", *, , =, %, "--", and new line characters <p>See Also: "Understanding Cross-Site Scripting Protection (page 19-21)"</p>

13.3.3.4 Creating a Quick Pick Selection

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%.

The screenshot shows a form with the following fields and their respective quick pick selections:

- Ename:
- Job:
- Mgr:
- Hiredate: (with a calendar icon)
- Sal:
- Comm: (with a quick pick selection showing 5%, 10%, 15%, and 20%)
- Deptno:

To create a quick pick:

1. Create the item. See "[Creating Page-Level Items in Component View](#) (page 13-22)."

2. View the item attributes. Under Items, select the item name.
The Edit Page Item page appears.
3. Scroll down to Quick Picks.
4. From Show Quick Picks, select **Yes**.
5. Define the Label and Value for each quick pick:
 - a. Under Label, enter the label to display for each selection.
 - b. Under Value, enter the corresponding value.
6. In Link Attributes, enter additional attributes for the quick picks.
7. Click **Apply Changes**.

13.3.3.5 Displaying Conditional Page Items

To display a conditional item:

1. Create the item. See "[Creating Page-Level Items in Component View](#) (page 13-22)."
2. View the item attributes. Under Items, select the item name.
The Edit Page Item page appears.
3. Scroll down to Conditions.
4. Make a selection from the Condition Type list.
5. Enter expressions in the fields provided.
6. Click **Apply Changes**.

13.3.3.6 Displaying Read Only Page Items

To display a read-only item:

1. Create the item. See "[Creating Page-Level Items in Component View](#) (page 13-22)."
2. View the item attributes. Under Items, select the item name.
The Edit Page Item page appears.
3. Scroll down to Read Only. Under Read Only:
 - a. Make a selection from Read Only Condition Type.
Depending on the item type, a read-only item may be rendered as plain text or as nonenterable field. Not selecting a condition causes the item to render normally.



Tip:

Setting Condition Type to **Never** always renders the item as an enterable field. This setting overwrites any read-only condition setting defined at the region or page-level.

- b. Enter expressions in the fields provided.

- c. In Read Only Element Table Cell(s) Attributes, enter attributes to be included in the HTML table cell tag, <td>, that contains the item when displayed in read only mode. For example:

```
bgcolor="#FF0000"
```

 **Tip:**

Read Only Element Table Cell(s) Attributes only be displays if the application uses HTML tables to layout regions and page items.

4. Click **Apply Changes**.

13.3.3.7 Editing Multiple Items Simultaneously

Item attributes control how items display on a page. You can use the Page Items page to edit the sequence, field label, template, region, and overall position for all items on a page. The sections that follow describe how to edit items using these pages.

- [Editing Multiple Items](#) (page 13-33)
- [Reassigning Region Items](#) (page 13-34)
- [Deleting Multiple Items](#) (page 13-35)
- [Viewing Item Utilities](#) (page 13-35)
- [Viewing Page Item History](#) (page 13-35)

13.3.3.7.1 Editing Multiple Items

To edit multiple items simultaneously:

1. Navigate to the page containing the items. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Page Items page. Under Items, click the **Edit All** icon. The Edit All icon resembles a small page with a pencil on top of it.

The Items tab appears.

A Navigation bar displays at the top of the page. Use the Page field to navigate to another page. Use the Name field to search for a specific item. To limit the display to just items in a specific region, make a selection from Regions list.

 **See Also:**

["Editing Page Item Attributes in Component View](#) (page 13-28)"

The Page Items page displays items in an editable report. [Table 13-4](#) (page 13-34) describes editable attributes.

Table 13-4 Editable Attributes on Page Items

Attribute	Description
Sequence	Specify the display sequence for this component. The sequence determines the order of evaluation.
Prompt	Enter the label for this HTML form element. You may include HTML, JavaScript, and shortcuts. You can also use the substitution string #CURRENT_ITEM_NAME# to obtain the name of the item associated with this label.
Field Template	Determines the label template. Label templates enable you to define the user interface attributes in a central place and share that definition among many labels.
Region	Defines the region in which the item displays. All items must be in a region.
New Line	Determines whether this item displays on the same line as the previous item or whether it displays on the next line. Items are laid out in an HTML table. Select Yes to have an item display as the first field in a new row in the table.
New Field	If set to Yes , this item is rendered into its own HTML table cell which is the default. If set to No , the item is rendered into the same HTML table cell as the previous page item. Note that rendering into the same table cell looks more attached to the previous page item.
Width	Specifies the length (in characters) of the form element that displays for this item.
Height	Specifies the height (in lines) for text areas and multi select lists.
Column Span	Items are laid out in HTML tables. This property defines the value to be used for the COLSPAN attribute in the table cell.
Row Span	Items are laid out in HTML tables. The attribute determines the value to be used for the ROWSPAN attribute in the table cell that the item displays in.
Post Element	Text or HTML code is rendered immediately after the page item element.

13.3.3.7.2 Reassigning Region Items

Use the Reassign Region Items page to assign items to a new region.

To assign items to a new region on the Reassign Region Items page:

1. Navigate to the page containing the items. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Page Items page. Under Items, click the **Edit All** icon. The Edit All icon resembles a small page with a pencil on top of it.
3. Click **Reassign Region Items**.

The Reassign Region Items page appears.

Use the Navigation bar at the top of the page to narrow or broaden the display. Available options include:

- **Page** - Select or enter a new number and click **Go**.

- **Show Region(s)** - Select a region to display and click **Go**.
4. To reassign an item to another region:
 - a. Select the items to reassign.
 - b. From Assign to Region, select a new region.
 - c. Click **Reassign**.

13.3.3.7.3 Deleting Multiple Items

Use the Delete Multiple Items page to simultaneously delete multiple items.

To delete multiple items:

1. Navigate to the page containing the items. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Page Items page. Under Items, click the **Edit All** icon. The Edit All icon resembles a small page with a pencil on top of it.
3. Click **Delete Multiple Items**.

The Delete Multiple Items page appears.

Use the Navigation bar at the top of the page to narrow or broaden the display. Use the Page field to navigate to a new page. Select or enter a new number and click **Go**.

4. To delete items:
 - a. Select the items.
 - b. Click **Remove Items**.

13.3.3.7.4 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. Navigate to the page containing the items. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Page Items page. Under Items, click the **Edit All** icon. The Edit All icon resembles a small page with a pencil on top of it.
3. Click **Utilities**.
4. 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
 - Conditional Items

13.3.3.7.5 Viewing Page Item History

Use the History page to view a summary of recent edits to page-level items.

To access the History page:

1. Navigate to the page containing the items. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Page Items page. Under Items, click the **Edit All** icon. The Edit All icon resembles a small page with a pencil on top of it.
3. Click **History**.

The Page Item History page appears.

13.4 Managing Dynamic Actions

Dynamic actions enable developers to define complex client-side behavior declaratively without the need for JavaScript.

- [About Dynamic Actions](#) (page 13-36)
- [Viewing Dynamic Action Examples](#) (page 13-37)
- [About Dynamic Action Events](#) (page 13-37)
- [Managing Dynamic Actions in Page Designer](#) (page 13-40)
- [Managing Dynamic Actions in Legacy Component View](#) (page 13-45)
- [About Calling JavaScript Using a Dynamic Action](#) (page 13-50)
- [Debugging Dynamic Actions](#) (page 13-50)



See Also:

["Accessing Page Specific Utilities](#) (page 6-79)"

13.4.1 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. See "[Debugging Dynamic Actions](#) (page 13-50)" for information on how to debug problems.

The process of implementing a dynamic action involves the following:

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.

**See Also:**

"[Debugging Dynamic Actions](#) (page 13-50)"

13.4.2 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 Packaged Application](#) (page 3-4)"

13.4.3 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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- **Click** (`click`) - Fires when the pointing device button is clicked over the triggering element. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- **Double Click** (`dblclick`) - Fires when the pointing device button is double clicked over the triggering element. Supported User Interface types: Desktop.
- **Get Focus** (`focus`) - Fires when the triggering element receives focus by either a pointing device or by tabbing into the element. Supported User Interface types: Desktop and jQuery Mobile Smartphone.

- 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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- 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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- 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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- Lose Focus (`focusout`) - Fires when the triggering element loses focus either by the pointing device or by tabbing out of the element. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- Mouse Button Press (`mousedown`) - Fires when the pointing device button is pressed over the triggering element. Supported User Interface types: Desktop.
- Mouse Button Release (`mouseup`) - Fires when the pointing device button is released over the triggering element. Supported User Interface types: Desktop.
- Mouse Enter (`mouseenter`) - Fires once when the pointing device is moved into the triggering element. Supported User Interface types: Desktop.
- Mouse Leave (`mouseleave`) - Fires once when the pointing device is moved away from the triggering element. Supported User Interface types: Desktop.
- Mouse Move (`mousemove`) - Fires when the pointing device is moved while it is over the triggering element. Supported User Interface types: Desktop.
- Orientation Change (`orientationchange`) - Fires when a device's orientation changes (by turning it horizontally or vertically). The 'this.browserEvent' JavaScript object contains an 'orientation' property that will equal either 'portrait' or 'landscape'. Supported User Interface Type: 'jQuery Mobile Smartphone'.
- Page Load (Desktop User Interface uses `ready`, jQuery Mobile Smartphone User Interface uses `pageinit`) – Fires when the page loads. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- Page Unload (`unload`) - Fires when a page is unloaded. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- Resize (`resize`) - Fires when the browser window is resized. Supported User Interface types: Desktop.
- 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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- 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). Supported User Interface types: Desktop and jQuery Mobile Smartphone.

- Scroll Start (`scrollstart`) - Fires when a scroll begins. Supported User Interface Type: jQuery Mobile Smartphone.
- Scroll Stop (`scrollstop`) - Fires when a scroll ends. Supported User Interface Type: jQuery Mobile Smartphone.
- Select (`select`) - Fires when a user selects some text in a text field. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- Swipe (`swipe`) - Fires when a horizontal drag of 30px or more (and less than 75px vertically) occurs, within a 1 second duration. Supported User Interface Type: jQuery Mobile Smartphone.
- Swipe Left (`swipeleft`) - Fires when a swipe event occurs moving in the left direction. Supported User. Interface Type: jQuery Mobile Smartphone.
- Swipe Right (`swiperight`) - Fires when a swipe event occurs moving in the right direction. Supported User Interface Type: jQuery Mobile Smartphone.
- Tap (`tap`) - Fires after a quick touch event. Supported User Interface Type: jQuery Mobile Smartphone.
- Tap and Hold (`taphold`) - Fires after a held touch event (held for close to 1 second). Supported User Interface Type: jQuery Mobile Smartphone.
- Touch Cancel (`vmousecancel`) - Normalized event for handling touch or mousecancel events. Supported User Interface Type: jQuery Mobile Smartphone.
- Touch End (`vmouseup`) - Normalized event for handling touchend or mouseup events. Supported User Interface Type: jQuery Mobile Smartphone.
- Touch Move (`vmousemove`) - Normalized event for handling touchmove or mousemove events. Supported User Interface Type: jQuery Mobile Smartphone.
- Touch Start (`vmousedown`) - Normalized event for handling touchstart or mousedown events. Supported User Interface Type: jQuery Mobile Smartphone.
- Virtual Click (`vclick`) - Normalized event for handling touchend or mouse click events. On touch devices, this event is dispatched after Touch End. Supported User Interface Type: jQuery Mobile Smartphone.

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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- Before Page Submit (`apexbeforepagesubmit`) - Fires before a page being submitted. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- 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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- 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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.

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. Supported User Interface types: Desktop and jQuery Mobile Smartphone.
- 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. Supported User Interface types: Determined by the plug-in's User Interfaces > Supported for settings.
- 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.

13.4.4 Managing Dynamic Actions in Page Designer

This section describes how to manage dynamic actions in Page Designer.

- [Creating a Dynamic Action in Page Designer](#) (page 13-40)
- [Editing Dynamic Actions in Page Designer](#) (page 13-43)
- [Defining Dynamic Action Event Scope](#) (page 13-44)
- [Deleting a Dynamic Action in Page Designer](#) (page 13-44)

13.4.4.1 Creating a Dynamic Action in Page Designer

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 \(page 13-37\)"](#)

13.4.4.2 Editing Dynamic Actions in Page Designer

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**.

See Also:

- ["Creating a Dynamic Action in Component View \(page 13-45\)"](#)
- ["Adding Additional True Actions \(page 13-48\)"](#)
- ["Defining the Frequency and Scope \(page 13-49\)"](#)

13.4.4.3 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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for a group or an attribute, enter a keyword in the Filter Properties field.

4. To view all attributes in the Property Editor, click **Show 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**.

13.4.4.4 Deleting a Dynamic Action in Page Designer

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**.

13.4.5 Managing Dynamic Actions in Legacy Component View

Developers can create and edit dynamic actions in Component View.



See Also:

["Managing Pages in Legacy Component View \(page 24-1\)"](#)

- [Creating a Dynamic Action in Component View \(page 13-45\)](#)
- [Editing Dynamic Actions in Component View \(page 13-47\)](#)
- [Adding Additional True Actions \(page 13-48\)](#)
- [Defining the Frequency and Scope \(page 13-49\)](#)

13.4.5.1 Creating a Dynamic Action in Component View

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.

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)."
2. Under Page Rendering, scroll down to Dynamic Actions and click the **Create** icon. The Dynamic Action wizard appears.
3. For Identification, enter the following and click **Next**.
 - Name - Enter the name of the dynamic action.
 - Sequence - Enter the sequence of this component. The sequence determines the order of execution.
4. For When, enter the following and click **Next**.
 - a. Event - Specify the event that will cause the dynamic action to fire. See "[About Dynamic Action Events \(page 13-37\)](#)."
 - b. Selection Type - Select the type of page element that triggers the dynamic action. A corresponding name field displays to specify the name of the page element.



Note:

Only available if the event selected supports definition of a page element. Selecting any of the following events hides this field: Page Load, Page Unload, Resize, Before Page Submit, Orientation Change. All other event types show this field.

- c. Specify the name of one or more page elements that will trigger the dynamic action. For multiples, separate page element names with a comma.
 - Item(s) - Enter or select one or more page item names that trigger the dynamic action. For multiple items, separate page items with a comma.
 - Button - Select the button that triggers the dynamic action. If the button is a Template button, the button selected must use a button template that includes an ID attribute set to #BUTTON_ID#. The dynamic action framework relies on this ID value to reference the button.
 - Region - Select the region name that triggers the dynamic action. The region selected must use a region template that includes a container element with an ID attribute set to #REGION_STATIC_ID#. The dynamic action framework relies on this ID value to reference the region.

The region can fire the event itself, for example, Before Refresh or After Refresh events, or can be a container to catch events that fire on items or other elements in the region. Only the following subset of events support this: Change (`change`), Click (`click`), Double Click (`dblclick`), Get Focus (`focus`), Key Down (`keydown`), Key Press, Key Release (`keyup`), Lose Focus (`blur`), Mouse Button Press (`mousedown`), Mouse Button Release (`mouseup`), Mouse Move (`mousemove`).
 - jQuery Selector - Enter the jQuery selector syntax to return one or more page elements that trigger the dynamic action.
 - JavaScript Expression - Enter a JavaScript expression to return a DOM or jQuery object that triggers the dynamic action.
 - d. Condition - To provide a condition, make a selection from the Condition list. Depending on the condition type, you may be required to enter a Value to test the condition against. If no condition is specified, only True Actions fire. If a condition is specified, the True Action fires when the condition is met, the False Action fires when it is not.
5. On True Action, specify the action performed when the previously specified event occurs and conditions are satisfied and click **Next**.
 - Action - Select the True action you want to fire when the Condition is met or when No Condition has been specified. Depending on the selected action, additional options display.

**Tip:**

To learn more, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

- Fire On Page Load - Select this option to trigger this action when the page loads. Depending upon the type of selected action, this option may or may not be selected.
- Generate Opposite False Action - Select this option to also generate the opposite false action. Actions treated as opposite are Show / Hide, Enable / Disable and Add Class / Remove Class. This option only displays if one of the above actions are selected, and a When Condition has been specified.

6. For Affected Elements (available only for actions that support affected elements), specify the elements affected by this dynamic action and how they are affected then click **Next**.

The final step in creating the dynamic action is different depending on whether the action you have selected supports selection of affected elements. Some actions, such as Submit Page, do not require affected elements to be selected. Other actions, such as Show, do. If the action selected does not support the selection of affected elements, the last page will be a Confirm page where you can review details of the dynamic action. If the action selected does support selection of affected elements, options on this page include:

- Selection Type - Select the type of page element that is affected by the dynamic action. Depending on the selected type, additional options display. Use these options to specify additional settings and values required to specify how the element is affected.

 **Tip:**

To learn more, make a selection and see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

7. Click **Create Dynamic Action**.

13.4.5.2 Editing Dynamic Actions in Component View

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. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Page Rendering, scroll down to Dynamic Actions and click the dynamic action you want to modify.

The Edit Dynamic Action page appears.

3. Edit the appropriate attributes.

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

4. Click **Apply Changes**.

 **See Also:**

"[Creating a Dynamic Action in Component View](#) (page 13-45)," "[Adding Additional True Actions](#) (page 13-48)," and "[Defining the Frequency and Scope](#) (page 13-49)"

13.4.5.3 Adding Additional True Actions

This section describes how to add additional true actions.

- [About Adding Additional True Actions](#) (page 13-48)
- [Adding a True Action](#) (page 13-48)

13.4.5.3.1 About Adding Additional True Actions

Dynamic actions contain a condition that can be specified to control when True actions fire. For example, consider an application that displays employee information to managers. If the employee is exempt, their salary displays, otherwise their hourly wage displays.

The Dynamic Action creation wizard allows only one true action to be specified. To add more than one action you must use the create wizard to create the dynamic action and then edit the action to add additional true actions.

13.4.5.3.2 Adding a True Action

To add an action:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Page Rendering, scroll down to Dynamic Actions and click the dynamic action you want to modify.

The Edit Dynamic Action page appears.

3. To add an action that executes when the conditions are met or when no condition is specified, scroll down to True Actions and click **Add True Action**.

The Create/Edit Action page appears.

4. Under Identification:
 - a. Sequence - Enter the sequence for this component. This indicates the execution order.
 - b. Action - Select the action that triggers when the condition is true.
5. Under Execution Options:
 - a. Fire on Page Load - Select if you want the action to also trigger when the page loads.
 - b. Edit the appropriate attributes.

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

6. For Affected Elements (only available if the selected action supports definition of affected elements):

 **Note:**

Some or all of these fields may not be available to select, depending on the type of action selected. For example the Disable action does not support selecting a Region as the affected element.

- a. Selection Type - Select the type of page element that is affected by the dynamic action. A corresponding name field displays to specify the name of the page element.
7. For Comments, enter developer comments or notes. These comments never display when the application is running.
8. Click **Create**.
The Edit Dynamic Action page displays with the added action listed under True Actions.

13.4.5.4 Defining the Frequency and 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. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Page Rendering, scroll down to Dynamic Actions and click the dynamic action you want to modify.
The Edit Dynamic Action page appears.
3. Scroll down to Advanced. For Event Scope, select one of the following:
 - a. Static (default) - Binds the event handler to the triggering elements for the lifetime of the current page, but will no longer be bound if the triggering elements are updated by Partial Page Refresh (PPR).
 - b. Dynamic - Binds the event handler to the triggering elements for the lifetime of the current page, including any elements that are updated by Partial Page Refresh (PPR). Specifying Dynamic causes an additional field Static Container (jQuery Selector) to be displayed. Specifying a Static Container can help improve the performance of the way events are handled with a Dynamic Event Scope. This should be an element on the page that itself does not get re-created, but contains the triggering elements that are re-created using Partial Page Refresh (PPR). This element reference must be defined as a jQuery selector. For example if you have a dynamic action that does something to the rows of an interactive report region (which is re-created by PPR), this would need an Event Scope of Dynamic, in order for the dynamic action to still work after the report has been refreshed. And here, the Static Container value could be set to a jQuery selector selecting the region's Static ID value, for example: '#my_region'.
 - c. Once - Binds the event handler to the triggering elements for a once only event.

4. Click **Apply Changes**.

13.4.6 About Calling JavaScript Using a Dynamic Action

You can also execute JavaScript code by creating a dynamic action. JavaScript code can be executed with the types **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 \(page 13-36\)"](#)

13.4.7 Debugging Dynamic Actions

This section describes how to debug dynamic actions in Oracle Application Express.

- [About Debugging Dynamic Actions \(page 13-50\)](#)
- [Debugging Dynamic Actions \(page 13-50\)](#)

13.4.7.1 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.

13.4.7.2 Debugging Dynamic Actions

To debug a dynamic action:

1. Ensure the application containing the dynamic action has Debugging enabled. See ["Utilizing Debug Mode \(page 22-4\)."](#)
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.

13.5 Creating Buttons

As you design your application, 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).

- [What Actions Can a Button Perform?](#) (page 13-51)
- [Managing Buttons in Page Designer](#) (page 13-51)
- [Managing Buttons in Legacy Component View](#) (page 13-55)
- [About Calling JavaScript from a Button](#) (page 13-60)
- [About the Relationship Between Button Names and REQUEST](#) (page 13-61)
- [About Branching with Buttons](#) (page 13-61)

See Also:

"[Calling a Page from a Button URL](#) (page 2-20)" and "[Accessing Page Specific Utilities](#) (page 6-79)"

13.5.1 What Actions Can a Button Perform?

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).

13.5.2 Managing Buttons in Page Designer

This section describes how to manage buttons in Page Designer.

- [Creating a Button in Page Designer](#) (page 13-52)
- [Editing a Button in Page Designer](#) (page 13-53)
- [Displaying a Button Conditionally](#) (page 13-53)
- [Configuring Button Attributes to Warn Users of Unsaved Changes](#) (page 13-54)

**See Also:**

["Managing Buttons in Legacy Component View \(page 13-55\)"](#)

13.5.2.1 Creating a Button 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.

5. To view all attributes in the Property Editor, click **Show All**.
6. To search for a group or an attribute, enter a keyword in the Filter Properties field.
7. 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.

8. Click **Save**.

**See Also:**

["About Regions \(page 12-4\)"](#)

13.5.2.2 Editing a Button 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.

Property Editor organizes attributes into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

3. To search for an attribute, enter a keyword in the Filter Properties field.

The attribute appears.

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**.

13.5.2.3 Displaying a Button Conditionally

You can choose to have a button display conditionally by editing attributes on the Edit Pages Button page.

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 into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

3. To search for a group or an attribute, enter a keyword in the Filter Properties field.

4. Locate and expand **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 \(page 2-24\)"](#)

13.5.2.4 Configuring Button Attributes to Warn Users of Unsaved Changes

Developers can use the **Warn on Unsaved Changes** button attribute to warn users of unsaved changes when they attempt to navigate away from the page by clicking the button.

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.
3. Locate and expand **Behavior**.

 **Tip:**

To search for an attribute, enter a keyword in the Filter Properties field. The attribute appears.

4. 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.
5. 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 \(page 7-36\)"](#) and *Oracle Application Express API Reference*

13.5.3 Managing Buttons in Legacy Component View

Developers create and edit buttons in Component View.

- [Creating a Button in Component View \(page 13-55\)](#)
- [Creating Multiple Buttons in Component View \(page 13-57\)](#)
- [Editing a Button in Legacy Component View \(page 13-58\)](#)
- [Displaying a Button Conditionally \(page 13-59\)](#)

 **See Also:**

[Managing Pages in Legacy Component View \(page 24-1\)](#)

13.5.3.1 Creating a Button in Component View

To create a region button in Component View:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)."
2. If necessary, create a region. See "[About Regions \(page 12-4\)](#)."
3. Under Buttons, click the **Create** icon.
4. On Button Region, select a region for the button and click **Next**.
5. On Button Position:
 - a. Position - Select a position for the button:
 - **Create a button in a region position** - Select this option to place the button in a region position. A region position is a position defined by a region template.
 - **Create a button displayed among this region's items** - Select this option to display the button within or between page items (for example, to add a button directly to the right of a form field).
 - b. Click **Next**.

6. For Button Attributes:

 **Tip:**

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

- a. **Button Name** - Enter a name for this button. For buttons with an action of **Submit Page**, the button name becomes the page request value. For example, if you name the button `FLOW_RESET_BUTTON`, then the button has purpose of resetting the current page without doing a submit).
- b. **Label** - Enter a label to be displayed for the button.
- c. **Button Template** - Define which template controls the display of the button. Button templates are defined on the templates page. You must have at least one button template defined to use button templates.
- d. **Button Type** - Use this attribute to select either the **Normal** or **Hot** template of the button template.

If you select **Hot**, the button is rendered using the template stored in the Hot Template attribute of a button template. If no value is defined, the regular button template is used.

- e. **Button Attributes** - Text entered is added to the HTML element definition. The following example sets the class of a text button:

```
class="myclass"
```

For buttons based on templates, your template must include the `#BUTTON_ATTRIBUTES#` substitution string. You can use this to control tab stops. For example:

```
tabindex="3"
```

- f. **Click Next.**

Display Properties appears. The options that display for Display Properties depend upon the selected region position.

7. For Display Properties:

- If the button position is **Create a button in a region position:**
 - **Sequence** - Enter the sequence for this component. The sequence determines the order of evaluation.
 - **Position** - Select a position for this button.
 - **Alignment** - Select **Right** or **Left**.
 - **Click Next.**
- If the button position is **Create a button displayed among this region's items:**
 - **Sequence** - Enter the sequence for this component. The sequence determines the order of evaluation.
 - **Click Next.**

8. For Action When Clicked:

Tip:

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

Action - Select a behavior when the button is clicked.

- b.** Execute Validations - Execute Validations controls the execution of validations when a page is submitted. For example, this is useful for a Delete button where you do not want to annoy the user with irrelevant validation error messages when they are deleting a record.

To have a specific validation always execute independent of this setting, set the validation attribute, Always Execute, to **Yes**. To learn more, see "[About Determining When Validations Execute](#) (page 15-19)."

- c.** Click **Next**.

9. For Conditional Display, select a condition type from the list that must be met in order for this component to be rendered.

10. Click **Create Button**.

See Also:

"[About the Relationship Between Button Names and REQUEST](#) (page 13-61)"

13.5.3.2 Creating Multiple Buttons in Component View

In Component View, you can create multiple Submit Page buttons within the same region at once using the Create Multiple Buttons Wizard.

To create multiple buttons at once:

- 1.** Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
- 2.** If necessary, create a region. See "[About Regions](#) (page 12-4)."
- 3.** In Component View, locate the Buttons section and click the **Create** icon.
The Create Button wizard appears.
- 4.** Click the **Create Multiple Buttons** link.
The Create Multiple Button Wizard appears.
- 5.** From Region, select the region to contain the buttons.
- 6.** From Template, select a template. If no template is selected, buttons of style HTML are created.

7. In HTML Attributes, specify HTML attributes for these buttons. This text is added to the HTML element definition. For example, you could set the class of a text button as follows:

```
class="myclass"
```

8. To quickly populate the remaining fields, make a selection from the Quick Button list on the right side of the page.
9. Click **Create Buttons**.

13.5.3.3 Editing a Button in Legacy Component View

Developers can edit a button in Component View.

- [Editing Button Attributes](#) (page 13-58)
- [About Execute Validations](#) (page 13-58)
- [Editing Multiple Buttons at Once](#) (page 13-59)

13.5.3.3.1 Editing Button Attributes

To edit attributes for an existing button:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Button, select the button name.
3. Edit the button attributes.
To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."
4. Click **Apply Changes**.

13.5.3.3.2 About Execute Validations

The Action When Button Click, Execute Validations attribute on the Edit Page Buttons page controls the execution of validations when a page is submitted. For example, this is useful for a Delete button where you do not want to annoy the user with irrelevant validation error messages when they are deleting a record.

For Execute Validations, you can select:

- **Yes** - If the page is submitted and this attribute is set to **Yes**, all validations defined for the page and all built-in validations execute (for example to check for a required value or to determine if the value is a valid number or date).
- **No** - If set to **No**, none of the defined validations or the built-in validations execute.

To have a specific validation always execute independent of this setting, set the validation attribute, Always Execute, to **Yes**.



See Also:

["About Determining When Validations Execute](#) (page 15-19)"

13.5.3.3.3 Editing Multiple Buttons at Once

Use the Buttons page to delete multiple buttons or view a history of recent changes.

To edit multiple buttons at once:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Buttons, click the **Edit All** icon. The Edit All icon resembles a small grid with a pencil on top of it.

The Buttons page appears. The Buttons page contains a table listing the buttons in the region.

3. Edit the attributes and click **Apply Changes**.
 - [Deleting Multiple Buttons](#) (page 13-59)
 - [History](#) (page 13-59)

13.5.3.3.3.1 Deleting Multiple Buttons

To delete multiple buttons at once:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Buttons, click the **Edit All** icon. The Edit All icon resembles a small grid with a pencil on top of it.

The Buttons page appears. The Buttons page contains a table listing the buttons in the region.

3. Click **Delete Multiple Buttons**.

The Delete Multiple Buttons page appears.
4. Select the buttons to delete and click **Remove Buttons**.

13.5.3.3.3.2 History

Use the History page to view a summary of recent edits to buttons.

13.5.3.4 Displaying a Button Conditionally

You can choose to have a button display conditionally by editing attributes on the Edit Pages Button page.

To have a button display conditionally:

1. Create the button. See "[Creating a Button in Component View](#) (page 13-55)".
2. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
3. Under Buttons, click the button name.
4. Scroll down to Conditions.
5. Make a selection from the Condition Type list.
6. Enter an expression in the fields provided.

7. Click **Apply Changes**.



See Also:

["Referencing Session State Using Bind Variable Syntax \(page 2-24\)"](#)

13.5.4 About Calling JavaScript from a Button

Calling JavaScript from a button is a great way 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**. See ["Creating a Button in Component View \(page 13-55\)"](#).
2. 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. To learn more, see ["Managing Dynamic Actions \(page 13-36\)"](#).

13.5.5 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 submits. 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.

13.5.6 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 \(page 14-25\)"](#)

13.6 Creating Trees

Trees to display hierarchical information in a clear, easy-to-use format. You can create a tree control using a SQL query.



Tip:

In Oracle Application Express release 5.0, you can continue to edit existing jsTree regions, but no longer have the ability to create them. See ["About Trees \(page 13-62\)"](#).

- [About Trees \(page 13-62\)](#)

- [About Support for Legacy jsTrees](#) (page 13-62)
- [Managing Trees in Page Designer](#) (page 13-62)
- [Managing Trees in Legacy Component View](#) (page 13-68)

**See Also:**

To view tree examples, install the packaged application, *Sample Trees*.

See "[Managing Packaged Applications in a Full Development Environment](#) (page 3-2)."

13.6.1 About Trees

App Builder includes a built-in wizard for generating a tree that displays a hierarchical view. With Oracle Application Express release 5.0, trees use a new APEX Tree implementation. Like the legacy jsTree region, this new tree implementation is a JavaScript-based, cross browser tree component that features optional keyboard navigation, and optional state saving. The APEX Tree shares the same functionality as jsTree, with the exception that the Template attribute does not apply to the new tree. The look, usability, and accessibility is also improved with the new implementation. You can continue to edit existing jsTree regions, but no longer have the ability to create them.

You can create a Tree from a query that specifies a hierarchical relationship by identifying an ID and parent ID column in a table or view. The tree query utilizes 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.

13.6.2 About Support for Legacy jsTrees

Previous releases of Oracle Application Express supported the creation of jsTree tree regions. Although existing trees are still supported, Oracle Application Express no longer supports the creation of 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, similar to jsTree.

13.6.3 Managing Trees in Page Designer

This section describes how to manage trees in Page Designer.

- [Creating a Tree on New Page](#) (page 13-63)
- [Creating a Tree in Page Designer](#) (page 13-65)
- [Editing Tree Attributes in Page Designer](#) (page 13-67)

13.6.3.1 Creating a Tree on New Page

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 this 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 **Desktop**.
 - b. Select a page type - Select **Tree**.
 - c. Click **Next**.
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, 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.
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.

- e. Click **Next**.
11. Confirm your selections and click **Create**.

 **Tip:**

The creation of APEX Tree regions is supported on pages using a Desktop user interface.

13.6.3.2 Creating a Tree in Page Designer

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.

Property Editor organizes attributes into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

To search for an attribute, enter a keyword in the Filter Properties field.
 5. In the Property Editor, edit these attributes:

- a. **Identification, Title** - Enter a region title. The region title only displays when it is defined in the region template.
- b. **Source, SQL Query** - Enter the SQL query underlying the tree. The query must be in the following format:

```
SELECT status, level, name, icon, id, tooltip, link
FROM ...
WHERE ...
START WITH...
CONNECT BY PRIOR id = pid

ORDER SIBLINGS BY ...
```

 **Note:**

The `STATUS` column of the tree SQL query is used by our tree logic in the rendering of the appropriate JSON data source for the tree, and should not be modified.

Consider the following example:

```
SELECT case WHEN connect_by_isleaf = 1 THEN 0 WHEN level = 1 THEN 1 ELSE -1
END as status,
       level,
       ename as title,
       'icon-tree-folder' as icon,
       empno as value,
       ename as tooltip,
       NULL as link
FROM emp
START WITH mgr IS NULL
CONNECT BY PRIOR empno = mgr
ORDER SIBLINGS BY ename
```

 **Note:**

By default, a new APEX Tree will be created, using CSS class 'icon-tree-folder' to represent the image on the tree. If you are creating a jsTree (Legacy) tree implementation, an image reference should be used in place of the CSS class in the above example, you would replace 'icon-tree-folder' with '#IMAGE_PREFIX#menu/go-small.png'.

- 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.
6. 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:
 - **Tree Implementation** - Select how the tree should be implemented, either **APEX Tree** or **jsTree (legacy)**.
 - **Tree Template** - Only displays for trees implemented with **jsTree (legacy)**. Select the template to apply to the tree from the available jsTree template options.
 - **Activate Node Link With** - Select the action required to activate node links on your tree.
 - **Selected Node Page Item** - Select the page or application item that you wish to hold the selected node value.
 - **Tooltip** - Select whether tooltips are displayed, and the source for the tooltip.
 - **Static Tree ID** - Enter the unique identifier for the tree.
This ID is only required for legacy JavaScript code in Buttons, to collapse and expand the trees. New applications should use Dynamic Actions to affect the tree.
 - **Icon Type** - A CSS class used to specify the type of icons used for all nodes. This option is only applicable to APEX Tree regions.
7. Click **Save**.

13.6.3.3 Editing Tree Attributes in Page Designer

By default, a tree does not have focus and node links are activated with a single click action. Developers can customize tree behavior by editing tree attributes.

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 for the tree in the right pane.

Property Editor organizes attributes into functional groups. To expand or collapse these groups, click the **Show Common**, **Show All**, **Collapse All**, and **Expand All** icons.

3. To search for a group or an attribute, enter a keyword in the Filter Properties field.
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**.

13.6.4 Managing Trees in Legacy Component View

 **Important:**

Legacy Component View is disabled by default. To enable Legacy Component View, edit your user preferences. See "[Editing User Preferences](#) (page 1-23)".

Legacy Component View is a legacy user interface for editing application pages. Oracle recommends developers use Page Designer instead of Legacy Component View. Legacy Component View groups user interface elements and application logic by component type. Unlike Page Designer, Legacy Component View does not offer a visual representation of a page or support the dragging and dropping of page components.

 **Tip:**

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. However, when you click a component and the Property Editor highlights the corresponding attribute in the right pane. See "[Component View Tab](#) (page 7-14)".

Developers can create and edit trees in Component View.

 **Note:**

[Managing Pages in Legacy Component View](#) (page 24-1)

- [Creating a Tree in Component View](#) (page 13-69)
- [Editing Tree Attributes in Component View](#) (page 13-70)

13.6.4.1 Creating a Tree in Component View

To create a tree in component view:

1. Navigate to the page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Region, click the **Create** icon.
The Create Region Wizard appears.
3. Select **Tree** and click **Next**.
4. For Display Attributes:
 - a. Title - Enter a title for the region. This title displays if the region template you choose displays the region title.
 - b. Region Template - Choose a template to control the look of the region.
 - c. Parent Region - Defines the parent region to which the new region belongs to.
 - d. Display Point - Identify a display point for this region. Two types of display points exist: **page template positions** and **page body positions**. To learn more about an attribute, see field-level Help.
 - e. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - f. Click **Next**.
5. 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**.
6. 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.

 **Tip:**

To learn more about an attribute, see field-level Help. See "[Viewing Field-Level Help](#) (page 1-27)."

3. Click **Apply Changes**.

13.7 About Incorporating JavaScript into an Application

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. Oracle Application Express includes multiple built-in interfaces especially designed for adding JavaScript.

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:**

"[Text with JavaScript Escaped Single Quotes](#) (page 16-43)" for information about referencing a shortcut inside of a JavaScript literal string, "[Understanding Validations](#) (page 15-8)," and "[About Using JavaScript Code in a Plug-In](#) (page 18-18)"

14

Adding Navigation

When you build a database application, you can include different types of navigation controls. The available navigation options depend upon the application theme. Common navigation controls include tabs, lists (including navigation menus), breadcrumbs, navigation bar entries, and branches. This section describes how to implement navigation in your application.

Topics:

- [About Creating Lists as Navigation](#) (page 14-1)
- [Creating Tabs](#) (page 14-2)
- [Creating Breadcrumbs](#) (page 14-8)
- [Creating a Navigation Bar Entry](#) (page 14-19)
- [Controlling Navigation Using Branches](#) (page 14-25)

See Also:

["Creating Trees](#) (page 13-61)"

14.1 About Creating Lists as Navigation

A list is 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 lists that function as navigation using the template Vertical Unordered List with Bullets. Newer themes, such as *Universal Theme - 42*, support the creation of navigation menus. Navigation menus are lists that render at the top of the page (similar to tabs) or display as a side bar. The Sample Database Application includes a navigation menu.

See Also:

- ["Creating Lists](#) (page 16-27)"
- ["About Navigation Menus](#) (page 16-28)"
- ["Understanding Sample Database Application](#) (page 3-8)"

14.2 Creating 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 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.

 **Note:**

When running the Create Application Wizard, you have the option of creating an application with tabs. The following procedures assume you have created an application that does *not* have any tabs.

- [About Tabs](#) (page 14-2)
- [About the Tabs Page](#) (page 14-3)
- [Checking for Tab Template Support](#) (page 14-4)
- [Creating a Tab](#) (page 14-5)
- [Editing Tabs](#) (page 14-7)
- [Accessing Tab Reports](#) (page 14-8)

 **See Also:**

- ["Creating Lists \(page 16-27\)"](#)
- ["About Navigation Menus \(page 16-28\)"](#)
- ["Creating a Database Application \(page 6-4\)"](#)

14.2.1 About Tabs

A standard tab can have a one-to-one relationship with a page, or a standard tab can be the current tab for many pages. 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 set 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. 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 packaged application, Sample Database Application.

 **See Also:**

["Understanding Sample Database Application \(page 3-8\)"](#)

14.2.2 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 \(page 14-4\)"](#)
- ["Understanding Sample Database Application \(page 3-8\)"](#)

14.2.3 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 \(page 11-59\)"](#)
- ["Using Themes \(page 11-1\)"](#) for information about setting a default page template at the application level

14.2.4 Creating a Tab

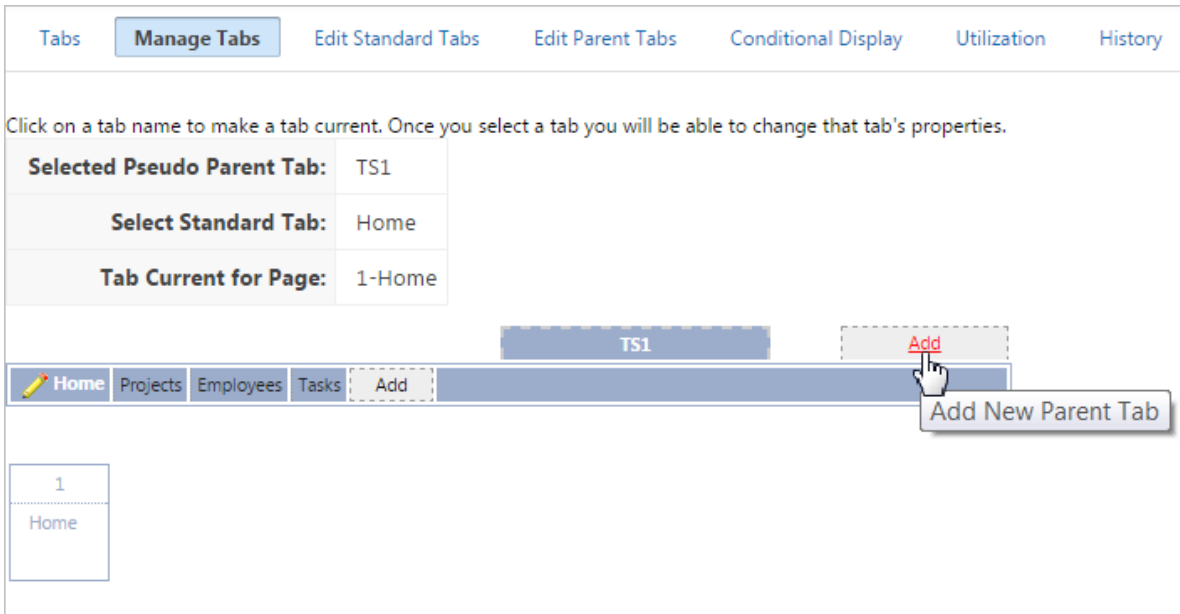
An application can have pages with no tabs, one level of tabs, and two levels of tabs. Standard tabs enable you to display only one level of tabs. To display two levels of tabs, you must define both Parent tabs and Standard tabs.

To create a tab:

1. Access the Tabs page:
 - a. On the Workspace home page, click the **App Builder** icon.
 - b. Select an application.

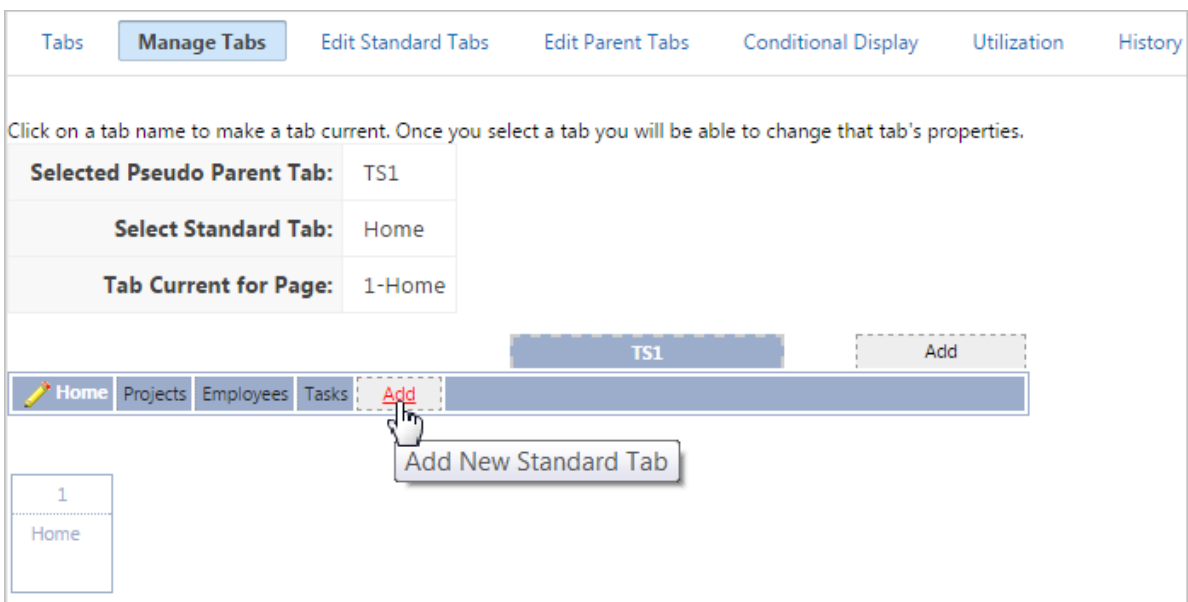
- c. On the Application home page, click **Shared Components**.
 - d. Under Navigation, click **Tabs**.
2. Click **Manage Tabs**.
 3. To add a Parent tab, click the **Add** button in the upper row.

A graphical representation of the tabs defined in your application displays.



Parent tabs can control the display of standard tab sets. Clicking a parent tab displays the corresponding standard tab, with the default page as the current page.

4. To create a Standard tab, click the **Add** button in the lower row.



The Create Parent Tab or Create Standard Tab Wizard appears.

5. Follow the on-screen instructions.



See Also:

["About Tabs \(page 14-2\)"](#)

14.2.5 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](#) (page 14-7)
- [Managing Standard Tabs](#) (page 14-7)

14.2.5.1 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**.

14.2.5.2 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**

14.2.6 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.

14.3 Creating 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.

- [What Are Breadcrumbs?](#) (page 14-9)
- [Creating a Breadcrumb While Creating a Page](#) (page 14-9)
- [Creating a Breadcrumb Manually](#) (page 14-10)
- [Editing Breadcrumbs](#) (page 14-16)
- [Reparenting Breadcrumb Entries](#) (page 14-17)

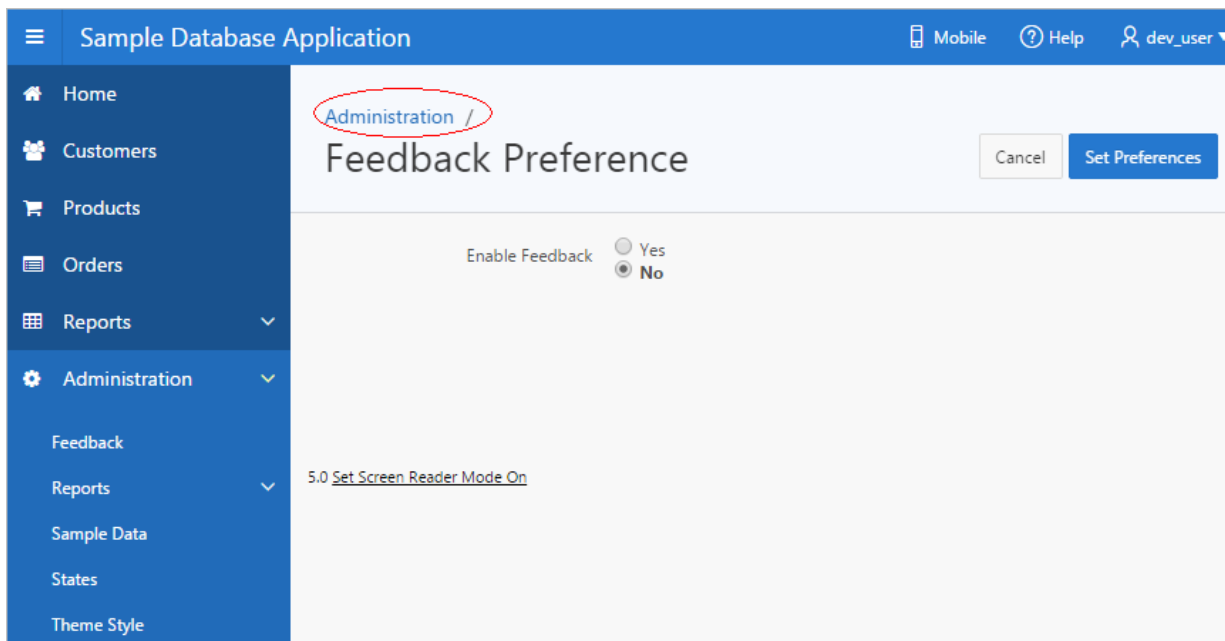
- [Deleting Unused Breadcrumb Entries](#) (page 14-17)
- [Accessing Breadcrumb Reports](#) (page 14-18)

 **See Also:**

- ["Creating a Theme](#) (page 11-20)"
- ["Breadcrumb Templates](#) (page 11-47)"

14.3.1 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](#) (page 3-8)"

14.3.2 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 \(page 6-44\)"](#)

14.3.3 Creating a Breadcrumb Manually

This section describes how to create breadcrumbs manually.

- [About Creating a Breadcrumb Manually \(page 14-10\)](#)
- [Creating Breadcrumbs \(page 14-10\)](#)
- [Adding Breadcrumb Entries \(page 14-12\)](#)
- [Adding Breadcrumb Regions \(page 14-13\)](#)
- [About Creating Dynamic Breadcrumbs \(page 14-16\)](#)

14.3.3.1 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 from the Shared Components page, Page Designer, or Component View.
2. Add entries to the breadcrumb.
3. Add the breadcrumb to a page by creating a region.

 **See Also:**

- ["Creating Breadcrumbs \(page 14-10\)"](#)
- ["Adding Breadcrumb Entries \(page 14-12\)"](#)
- ["Adding Breadcrumb Regions \(page 14-13\)"](#)

14.3.3.2 Creating Breadcrumbs

This section describes the various way to create breadcrumbs.

- [Creating Breadcrumbs as Shared Components](#) (page 14-11)
- [Creating a Breadcrumb Region in Page Designer](#) (page 14-11)
- [Creating Breadcrumbs from Component View](#) (page 14-12)

14.3.3.2.1 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 Breadcrumb Entries](#) (page 14-12)."

14.3.3.2.2 Creating a Breadcrumb Region in Page Designer

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**.

14.3.3.2.3 Creating Breadcrumbs from Component View

To create breadcrumbs from a Component View:

1. View the page in Component View. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Click the **Create** menu and select **Shared component**.
3. On Create Application Component, select **Breadcrumb** and click **Next**.
4. In Breadcrumb name, enter a name for this breadcrumb and click **Create**.
5. Once you create a named breadcrumb, you add breadcrumb entries.



See Also:

["Adding Breadcrumb Entries](#) (page 14-12)"

14.3.3.3 Adding Breadcrumb Entries

To add a breadcrumb entry:

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.

- [Adding a Breadcrumb Region in Page Designer](#) (page 14-14)
- [Adding a Breadcrumb Region in Legacy Component View](#) (page 14-15)

 **See Also:**

- "[Creating a New Template](#) (page 11-35)"
- "[Breadcrumb Templates](#) (page 11-47)" for information about changing menu display

14.3.3.4.1 Adding a Breadcrumb Region in Page Designer

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.

 **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 view all attributes in the Property Editor, click **Show 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.

14.3.3.4.2 Adding a Breadcrumb Region in Legacy Component View

To create a breadcrumb region in Component View:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Regions, click the **Create** icon.
The Create Region Wizard appears.
3. For the region type, select **Breadcrumb** and click **Next**.
4. For Breadcrumb Container Region:
 - a. Region Title - Enter a title for this region.
 - b. Region Template - Select a region template.
 - c. Display Point - Select a Display Point.
Regions are organized by position (or Display Point). To determine the appropriate region position, click the flashlight icon. A graphical representation appears.
 - d. Sequence - Enter a number for the sequence. The sequence determines the order of evaluation.
 - e. Click **Next**.
5. For Breadcrumb:
 - a. Breadcrumb - Select the breadcrumb to be associated with this region.

6. Click **Apply Changes**.

See Also:

- "[Viewing Field-Level Help](#) (page 1-27)"
- "[Accessing Breadcrumb Reports](#) (page 14-18)"

14.3.5 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**.

14.3.6 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**.

14.3.7 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 \(page 8-92\)"](#)

14.4 Creating a Navigation Bar Entry

This section explains how to create and edit navigation bar entries. Navigation bar entries offer an easy way to move users between pages in an application. For example, a navigation entry enables you to display a link from an image or text. The associated page template determines the location of a navigation bar.

- [About Navigation Bars \(page 14-19\)](#)
- [Creating a Navigation Bar Entry \(page 14-20\)](#)
- [Copying a Navigation Bar Entry \(page 14-23\)](#)
- [Editing a Navigation Bar Entry \(page 14-23\)](#)
- [Editing Multiple Navigation Bar Entries Simultaneously \(page 14-24\)](#)
- [Accessing Navigation Bar Entry Reports \(page 14-24\)](#)

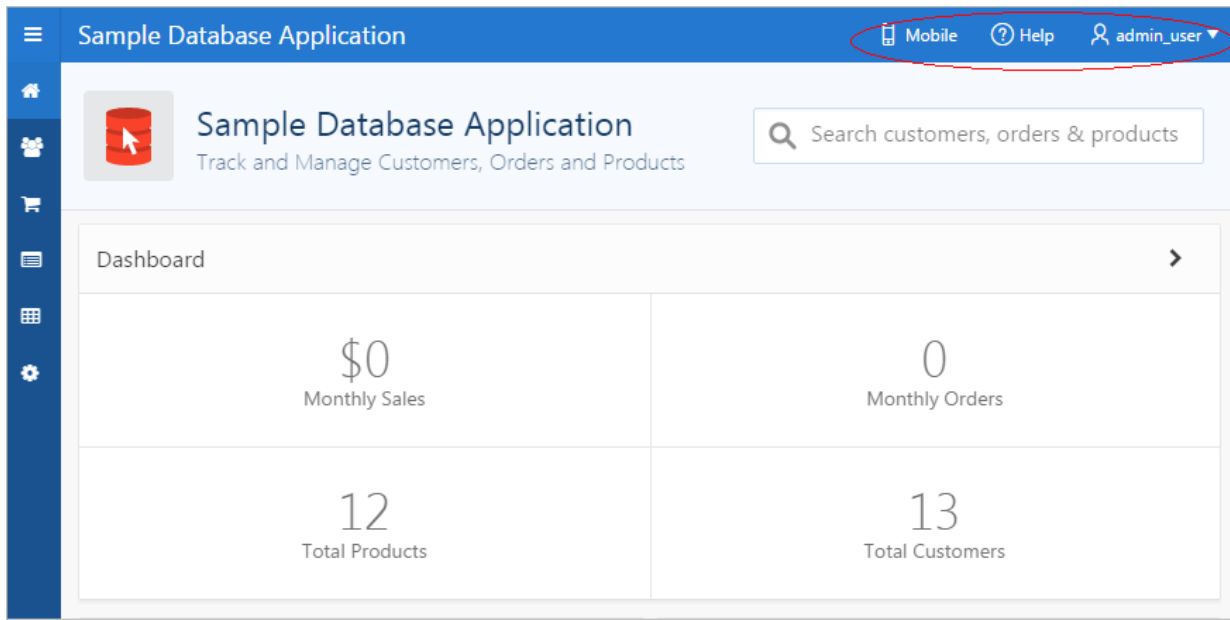


See Also:

["Creating Custom Themes \(page 11-32\)"](#)

14.4.1 About Navigation Bars

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.



See Also:

["Understanding Sample Database Application \(page 3-8\)"](#)

14.4.2 Creating a 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 Navigation Bar Entry from Scratch](#) (page 14-20)
- [Creating a Navigation Bar Entry for Feedback](#) (page 14-22)

14.4.2.1 Creating a 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 **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](#) (page 11-59)."

 **See Also:**

- "[Creating a Navigation Bar Entry for Feedback](#) (page 14-22)"
- "[Viewing Field-Level Help](#) (page 1-27)"
- "[Editing Themes](#) (page 11-10)"
- "[Optimizing a Page for Printing](#) (page 12-2)"

14.4.2.2 Creating a 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 **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.
 - b. Entry Label - Enter the label of the new navigation bar entry.
 - c. Display Sequence - The sequence for the new navigation bar entry. The sequence determines the order of evaluation.
6. Click **Create**.

 **Tip:**

In order for a navigation bar to appear, your page template must include the #NAVIGATION_BAR# substitution string. See "[Page Templates](#) (page 11-59)."

 **See Also:**

"[Managing Feedback](#) (page 4-20)"

14.4.3 Copying a Navigation Bar Entry

To copy 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 **Navigation Bar Entries**.
2. Click **Create**.

The Create Navigation Bar Entry Wizard appears.
3. For Method, select **As a Copy of an Existing Navigation Bar** and click **Next**.
4. For Copy From Application, select an application and click **Next**.
5. For New Names, enter the name on the new entry, and select whether to copy and subscribe.
6. Click **Copy Navigation Bar Entries**.

14.4.4 Editing a Navigation Bar Entry

Once you create a navigation bar entry, you can edit it on the Navigation Bar Entries page.

To edit a navigation bar entry:

1. Navigate to the Navigation Bar Entries page:
 - a. Navigate to the Workspace home 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 **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.

14.5 Controlling Navigation Using Branches

This section explains how to add a branch to an application. A branch is an instruction to link to a specific page, procedure, or URL after a given page is submitted. For example, you can branch from page 1 to page 2 after page 1 is submitted.

- [Creating a Branch in Page Designer](#) (page 14-25)
- [Creating a Branch in Component View](#) (page 14-27)
- [About Branching Conditionally](#) (page 14-28)



See Also:

["Accessing Page Specific Utilities](#) (page 6-79)"

14.5.1 Creating a Branch in Page Designer

To create a branch 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 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. To view all attributes in the Property Editor, click **Show All** .
 4. Edit the following 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.

5. Click **Save**.

 **See Also:**

- ["About the When Button Pressed Attribute \(page 2-44\)"](#)
- ["Creating Pre-Rendering Branches, Computations, and Processes \(page 7-30\)"](#)
- ["Creating Page Submission Branches, Validations, Computations, and Processes \(page 7-30\)"](#)

14.5.2 Creating a Branch in Component View

You create a branch by running the Create Branch Wizard and specifying the branch point and branch type.

To create a branch:

1. Navigate to the page. See ["Viewing a Page in Legacy Component View \(page 24-10\)"](#).
2. Under Branches, click the **Create** icon.
The Create Branch Wizard appears.
3. For Branch Attributes:
 - a. Name - Specify the name of the branch for easy identification by developers.
 - b. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - c. Branch Point - Select a branch point. Options include:
 - **On Submit: Before Computation** - Branching occurs before computations, validations, or processing. Use this option for buttons that should not invoke processing (for example, a Cancel button).
 - **On Submit: Before Validation** - Branching occurs after computations, but before validations or processing. If a validation fails, page processing stops, a rollback is issued, and the page displays the error. Because of this default behavior, you do not need to create branches to accommodate validations. However, you may want to branch based on the result of a computation (for example, to a previous branch point).
 - **On Submit: Before Processing** - Branching occurs after computations and validations, but before processing. Use this option to branch based on a validated session state, but before performing any page processing.
 - **On Submit: After Processing** - Branching occurs after computations, validations, and processing. This option branches to a URL or page after performing computations, validations, and processing. When using this option, remember to sequence your branches if you have multiple branches for a given branch point.
 - **On Load: Before Header** - Branching occurs before a page is rendered. This option displays another page instead of the current page or redirects the user to another URL or procedure.
 - d. Branch Type - Select a branch type. The most common branch type is **Branch to Page or URL**.

- e. Click **Next**.
4. For Target, follow the on-screen instructions. The attributes that display depend upon the selected **Branch Type**. To learn more, see item Help.
5. For Branch Conditions:
 - a. When Button Pressed - To execute this branch only when the user clicks a specific button, select a button from the list. You can incorporate this button condition with other conditions to further refine when this branch executes.
 - b. Condition Type - Select a condition type from the list that must be met in order for this component to be rendered or processed.
6. Click **Create Branch**.



See Also:

["About the When Button Pressed Attribute \(page 2-44\)"](#)

14.5.3 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.

15

Managing Computations, Validations, and Processes

This section describes how to manage computations, validations, and processes.

Topics:

- [Understanding Page Computations](#) (page 15-1)
- [Understanding Validations](#) (page 15-8)
- [Understanding Page Processes](#) (page 15-22)
- [About Branches](#) (page 15-32)
- [Understanding Application Processes](#) (page 15-32)
- [Understanding Application Computations](#) (page 15-37)
- [Using the Attribute Dictionary](#) (page 15-40)

See Also:

- ["Quick Start](#) (page 1-1)"
- [" App Builder Concepts](#) (page 2-1)"
- ["Creating Database Applications](#) (page 6-1)"
- ["Controlling Page Layout](#) (page 12-1)"
- ["Adding Navigation](#) (page 14-1)"

15.1 Understanding Page Computations

This section describes how to manage page computations in Page Designer and in Component View. You can use page computations to assign a value to an identified item when a page is submitted or displayed. 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.

- [About Page Computations](#) (page 15-2)
- [Managing Page Computations in Page Designer](#) (page 15-2)
- [Managing Page Computations in Legacy Component View](#) (page 15-4)
- [Computation Points and Computation Syntax Example](#) (page 15-6)



See Also:

["Understanding Application Computations \(page 15-37\)"](#)

15.1.1 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.

15.1.2 Managing Page Computations in Page Designer

This section describes how to create page computations in Page Designer.



Tip:

Learn more about execution behavior in the case of a validation error.

- [Creating a Page Computation in Page Designer \(page 15-2\)](#)
- [Editing Page Computation Attributes in Page Designer \(page 15-3\)](#)
- [About Editing the Computation Point and Source \(page 15-4\)](#)
- [About Creating Conditional Computations \(page 15-4\)](#)



See Also:

- ["Understanding Application Computations \(page 15-37\)"](#)
- ["What Happens When a Validation Fails? \(page 15-21\)"](#)

15.1.2.1 Creating a Page Computation in Page Designer

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**.
The Property Editor displays Computation attributes.
4. To view all attributes in the Property Editor, click **Show All**.
5. 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.

 **Tip:**

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

Next, define the action that to be performed if the event evaluates to True or False.

6. Click **Save**.

 **See Also:**

- ["Viewing a Page in Page Designer \(page 7-22\)"](#)
- ["Creating Pre-Rendering Branches, Computations, and Processes \(page 7-30\)"](#)
- ["Creating Page Submission Branches, Validations, Computations, and Processes \(page 7-30\)"](#)

15.1.2.2 Editing Page Computation Attributes in Page Designer

Once you create a computation, you can edit it on the Edit Page Computation page.

To edit a page computation:

1. View the page to contain the item in Page Designer.
2. In the Rendering tab, expand the Pre-Rendering or Post-Rendering node and select the computation.
3. In Property Editor - Computation, 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 \(page 7-22\)."](#)

15.1.2.3 About Editing the Computation Point and Source

You control when a computation executes using the Execution Options, Sequence and Execution Options, 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. For example, selecting **On New Instance** executes the computation when a new session (or instance) is generated.

After you specify the Sequence and Point, enter an expression or query under Computation to compute an item's value. In the event a computation fails, you can optionally define an error message in the Error Message attribute.



See Also:

["Editing Page Computation Attributes in Component View \(page 15-5\)"](#)

15.1.2.4 About Creating Conditional Computations

You can make a computation conditional by viewing the computation attributes in the Property Editor. Under Server-side Condition, edit the appropriate attributes.



See Also:

- ["Understanding Conditional Rendering and Processing \(page 2-10\)"](#)
- ["Accessing Property Editor \(page 7-24\)"](#)

15.1.3 Managing Page Computations in Legacy Component View

Developers can create and edit page computations in Component View.



Tip:

To learn more about execution behavior in the case of a validation error, see ["What Happens When a Validation Fails? \(page 15-21\)"](#)

- [Creating a Page Computation in Component View \(page 15-5\)](#)
- [Editing Page Computation Attributes in Component View \(page 15-5\)](#)

- [About Editing the Computation Point and Source](#) (page 15-6)
- [About Creating Conditional Computations](#) (page 15-6)

 **See Also:**

- ["Understanding Application Computations](#) (page 15-37)"
- ["Accessing Page Specific Utilities](#) (page 6-79)"

15.1.3.1 Creating a Page Computation in Component View

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 select a computation type.

To create a page computation:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Create Computation Wizard. Under Computations, click the **Create** icon.
3. For Item Type, select the location where the computation executes and click **Next**. Location options include:
 - a. Item on This Page
 - b. Item on Another Page
 - c. Application Level Item
4. For Item:
 - a. Compute Item - Select the item the computation updates.
 - b. Sequence - Select the order of evaluation.
 - c. Computation Point - Select the point at which the computation executes. The computation point **On New Instance** executes the computation when a new session (or instance) is generated.
 - d. Computation Type - Select the method of computation you want to create.
 - e. Click **Next**.
5. In Computation, enter a computation that corresponds to the selected computation type and click **Next**.
6. On Condition, you can choose to make the computation conditional. To make a computation conditional, make a selection from the Condition Type list and enter text in the expression fields.
7. Click **Create Computation**.

15.1.3.2 Editing Page Computation Attributes in Component View

Once you create a computation, you can edit it on the Edit Page Computation page.

To edit a page computation:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Edit Page Computation page. Under Computations, select the computation name.

The Edit Page Computation page appears.

3. Edit the appropriate attributes.
4. Click **Apply Changes**.

15.1.3.3 About Editing the Computation Point and Source

You control when a computation executes using the Sequence and Computation Point attributes on the Edit Page Computation page. The Sequence attribute determines the order of evaluation. The Computation Point attribute enables you to specify the point at which to execute the computation the item. For example, selecting **On New Instance** executes the computation when a new session (or instance) is generated.

After you specify the Sequence and Computation Point, under Source, enter an expression or query in the Computation field to compute an item's value. In the event a computation fails, you can optionally define an error message in the Computation Error Message field.



See Also:

"[Editing Page Computation Attributes in Component View](#) (page 15-5)"

15.1.3.4 About Creating Conditional Computations

You can make a computation conditional by making a selection from the Condition Type list and entering text in the expression fields on the Edit Page Computation page.



See Also:

- "[Editing Page Computation Attributes in Component View](#) (page 15-5)"
- "[Understanding Conditional Rendering and Processing](#) (page 2-10)"

15.1.4 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. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Create an item named `P10_PHONE_NUMBER` to store the combined values of `P10_AREA_CODE`, `P10_PREFIX`, and `P10_LINE_NUMBER`.
3. In Component view, under Computations, click the **Create** icon.
4. For Item Location, select **Item on this Page** and click **Next**.
5. For Computation Item, select **P10_PHONE_NUMBER**.
6. For Sequence, select the order of evaluation.
7. For Computation Type, you can create the following computation types:

a. Static Assignment:

- For Computation Type, select **Static Assignment** and click **Next**.

- Enter the following computation:

```
(&P10_AREA_CODE.) &P10_PREFIX.-&P10_LINE_NUMBER.
```

- Click **Next**.

b. PL/SQL Function Body:

- For Computation Type, select **PL/SQL Function Body** and click **Next**.

- Enter the following computation:

```
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;
```

- Click **Next**.

c. SQL Query (return colon separated value):

- For Computation Type, select **SQL Query (return colon separated value)** and click **Next**.

- Enter the following computation:

```
SELECT '('||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER FROM
DUAL
```

- Click **Next**.

d. PLSQL Expression:

- For Computation Type, select **PLSQL Expression** and click **Next**.

- Enter the following computation:

```
(''||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER
```

- Click **Next**.
8. Click **Create**.



See Also:

"About the Differences Between Page Items and Application Items
(page 13-2)"

15.2 Understanding Validations

This section describes how to create validations.

- [About Validations](#) (page 15-8)
- [Managing Validations in Page Designer](#) (page 15-8)
- [Managing Validations in Legacy Component View](#) (page 15-11)
- [About Determining When Validations Execute](#) (page 15-19)
- [About Defining How Validation Error Messages Display](#) (page 15-20)
- [Processing Validations Conditionally](#) (page 15-20)
- [About Altering the Go to Error Link](#) (page 15-21)
- [What Happens When a Validation Fails?](#) (page 15-21)

15.2.1 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.

15.2.2 Managing Validations in Page Designer

This section describes how to create validations.

- [Creating a Validation in Page Designer](#) (page 15-9)
- [Editing a Validation in Page Designer](#) (page 15-10)

 **See Also:**

 **See Also:**

- ["Validating User Input in Forms \(page 9-22\)"](#)
- ["About Execute Validations \(page 13-58\)"](#)
- ["About Item Types \(page A-1\)"](#)

15.2.2.1 Creating a Validation in Page Designer

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.
To view all attributes in the Property Editor, click **Show All**.
4. To search for a group or an attribute, enter a keyword in the Filter Properties field.
5. Edit the following attributes:
 - 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.

 **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:**

- ["Creating Pre-Rendering Branches, Computations, and Processes \(page 7-30\)"](#)
- ["Creating Page Submission Branches, Validations, Computations, and Processes \(page 7-30\)"](#)

15.2.2.2 Editing a Validation in Page Designer

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.
4. To view all attributes in the Property Editor, click **Show All**.
5. To search for a group or an attribute, enter a keyword in the Filter Properties field.
6. 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.

7. Click **Save**.

15.2.3 Managing Validations in Legacy Component View

Developers create a validation by running the Create Validation Wizard. This wizard changes slightly depending upon what you are validating. Validations are either specific to a given field (either a single page item or tabular form column validation) or, they are not field specific (either a page or tabular form row validation).

- [Creating a Validation for a Page Item](#) (page 15-11)
- [Creating a Validation for an Entire Page](#) (page 15-14)
- [Creating a Validation for a Tabular Form Column](#) (page 15-16)
- [Creating a Validation for a Tabular Form Row](#) (page 15-17)
- [Editing a Validation in Component View](#) (page 15-18)

 **See Also:**

- ["Validating User Input in Forms](#) (page 9-22)"
- ["About Execute Validations](#) (page 13-58)"
- ["About Item Types](#) (page A-1)"

15.2.3.1 Creating a Validation for a Page Item

To create a validation for a page item:

 **Note:**

Text entered for validations may not exceed 3,950 characters.

1. Navigate to the appropriate page. See ["Viewing a Page in Legacy Component View](#) (page 24-10)."

2. Access the Create Validation Wizard. Under Page Processing, locate Validations. Click the **Create** icon.
The Create Validation Wizard appears.
3. For Identify the validation level, select **Page Item** and click **Next**.
4. For Identify the Page Item that is to be validated, select the item to be validated and click **Next**.
5. For Sequence and Name:
 - a. Sequence - Enter the sequence number for this item.
 - b. Validation Name - Enter a name for the validation.
 - c. Error Display Location - Select the display location of the validation error message. Validation error messages can display on an error page or inline with the submitted page. Inline validations can display in the *notification* area (defined as part of the page template) or within the field label.
 - d. Click **Next**.
6. For Validation Type, select a method as described in [Table 15-1](#) (page 15-12) and click **Next**.

Table 15-1 Page Item Validation Methods

Validation Method	Descriptions
Not Null	<p>Checks if an item's value in session state is null.</p> <p>For example, you could validate that the user enters a value in a field by creating an item validation and then selecting the validation method Item Not Null.</p>
String Comparison	<p>Compares the value of an item to a specific string. When prompted select a validation type.</p> <p>There are several string comparison validations that compare the value of an item to a literal string. For example, you select the validation type Item in Expression 1 is contained in Expression 2 to validate a user entry in a field against a list of values you provide.</p> <p>In Expression 1, enter the name of the item you want to validate without a colon. For example:</p> <p>P1_VALUE</p> <p>In Expression 2, enter a string of values you want to validate against. For example:</p> <p>ABC/DEF/GHI</p>

Table 15-1 (Cont.) Page Item Validation Methods

Validation Method	Descriptions
Regular Expression	<p>Regular expressions provide a method to describe text patterns. Use a Regular Expression validation to perform data validation. When prompted, enter a regular expression.</p> <p>For example, you could use the following regular expression validation to verify that a string of entered data always consists of groups of six numbers separated by commas and followed by a comma:</p> <pre data-bbox="987 611 1227 636">^([[[:digit:]]{6},)+\$</pre> <p>This regular expression would find the following entries valid:</p> <pre data-bbox="987 737 1240 835">123456,654321, 123456, 123456,123456,654321,</pre> <p>However, the following would not be valid:</p> <pre data-bbox="987 884 1133 940">123456,12345 12345</pre>
SQL	<p>Compares to values in the database. Select a validation type:</p> <ul data-bbox="987 1035 1219 1129" style="list-style-type: none"> • Exists • NOT Exists • SQL Expression <p>For example, you can use a SQL validation to verify whether a last name typed into a field exists in the database. In the following Exists SQL validation, the field is named P1_LAST_NAME and the table is named customers.</p> <pre data-bbox="987 1329 1357 1381">SELECT 1 FROM customers WHERE last_name = :P1_LAST_NAME</pre> <p>To learn more, see the syntax examples at the bottom of the page.</p>

Table 15-1 (Cont.) Page Item Validation Methods

Validation Method	Descriptions
PL/SQL	<p>Useful if you need complex logic. Select a validation type:</p> <ul style="list-style-type: none"> • PL/SQL Expression • PL/SQL Error • Function Returning Boolean • Function Returning Error Text <p>For example, suppose you need to create a validation for an address form that requires the user to enter a province if the address is not in the United States. You could create the validation as a Function Returning Boolean, using the following PL/SQL:</p> <pre>BEGIN IF :P1_COUNTRY = 'US' AND :P1_PROVINCE IS NULL THEN RETURN FALSE; ELSE RETURN TRUE; END IF; END;</pre> <p>You could also create the same validation implemented as a PL/SQL Expression as follows:</p> <pre>NOT (:P1_COUNTRY='US' AND :P1_PROVINCE IS NULL);</pre> <p>To learn more, see the syntax examples at the bottom of the page.</p>

7. For Validation:
 - a. Error Message - Depending upon the validation method, enter the validation or message text that displays if the validation fails.
You can use #LABEL# as placeholder for the label of the associated page item.
 - b. Always Execute - Determines if validations execute when a page is submitted. Use this attribute with the Execute Validations attribute for buttons or certain page items that submit a page. Option include:
 - **Yes** - The validation always executes independent of validation settings for buttons or items on a page. Note that defined conditions still apply.
 - **No** - The validation only executes if the button or item used to submit the page has the **Execute Validations** attributes set to **Yes**.
 - c. Click **Next**.
8. Define conditions that apply to this validation and click **Create**.

15.2.3.2 Creating a Validation for an Entire Page

To create a validation for an entire page:

 **Note:**

Text entered for validations may not exceed 3,950 characters.

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. To access the Create Validation Wizard. Under Page Processing, locate Validations. Click the **Create** icon.
The Create Validation Wizard appears.
3. For Validation Level, select **Page** and click **Next**.
4. For Sequence and Name:
 - a. Sequence - Enter the sequence number for this item.
 - b. Validation Name - Enter a name for the validation.
 - c. Error Display Location - Select the display location of the validation error message. Validation error messages can display on an error page or inline with the submitted page. Inline validations can display in the *notification* area (defined as part of the page template) or within the field label.
 - d. Click **Next**.
5. For Validation Type, select a method as described in [Table 15-2](#) (page 15-15) and click **Next**.

Table 15-2 Page Validation Methods

Validation Method	Descriptions
SQL	<p>Compares to values in the database. Select a validation type:</p> <ul style="list-style-type: none"> • Exists • NOT Exists • SQL Expression <p>To learn more, see the syntax examples at the bottom of the page.</p>
PL/SQL	<p>Useful if you need complex logic. Select a validation type:</p> <ul style="list-style-type: none"> • PL/SQL Expression • PL/SQL Error • Function Returning Boolean • Function Returning Error Text <p>To learn more, see the syntax examples at the bottom of the page.</p>

6. For Validation:
 - a. Validation Code - Enter the code corresponding to the validation type selected.
 - b. Error Message - Depending upon the validation method, enter the validation or message text that displays if the validation fails.
You can use #LABEL# as placeholder for the label of the associated page item.

- c. Always Execute - Determines if validations execute when a page is submitted. Use this attribute with the Execute Validations attribute for buttons or certain page items that submit a page. Option include:
 - **Yes** - The validation always executes independent of validation settings for buttons or items on a page. Note that defined conditions still apply.
 - **No** - The validation only executes if the button or item used to submit the page has the **Execute Validations** attributes set to **Yes**.
 - d. Click **Next**.
7. Define conditions that apply to this validation and click **Create**.



See Also:

["About Determining When Validations Execute \(page 15-19\)"](#)

15.2.3.3 Creating a Validation for a Tabular Form Column

To create a validation for a tabular form column:



Note:

Text entered for validations may not exceed 3,950 characters.

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)."
2. To access the Create Validation Wizard. Under Page Processing, locate Validations. Click the **Create** icon.
The Create Validation Wizard appears.
3. For Validation Level:
 - a. Tabular Form - Select the associated tabular form region for this Validation.
 - b. Identify the validation level - Select **Column**.
 - c. Click **Next**.
4. For Identify the Column that is to be validated, select a column and click **Next**.
5. For Sequence and Name specify the following and click **Next**.
 - Sequence - Enter the sequence number for this item.
 - Validation Name - Enter a name for the validation.
 - Error Display Location - Select the display location of the validation error message. Validation error messages can display on an error page or inline with the submitted page. Inline validations can display in the *notification* area (defined as part of the page template) or within the field label.
6. For Validation Type, select a method, follow the on-screen instructions and click **Next**.

7. For Validation:
 - a. Define the validation. The attributes that display depend upon the Validation Type selected.
 - b. Error Message - Depending upon the validation method, enter the validation or message text that displays if the validation fails.

You can use #COLUMN_HEADER# as placeholder for the column header of the associated tabular form column.
 - c. Always Execute - Determines if validations execute when a page is submitted. Use this attribute with the Execute Validations attribute for buttons or certain page items that submit a page. Option include:
 - **Yes** - The validation always executes independent of validation settings for buttons or items on a page. Note that defined conditions still apply.
 - **No** - The validation only executes if the button or item used to submit the page has the **Execute Validations** attributes set to **Yes**.
 - d. Click **Next**.
8. For Conditions, define conditions that apply to this validation.
9. Click **Create Validation**.

 **See Also:**

["About Determining When Validations Execute \(page 15-19\)"](#)

15.2.3.4 Creating a Validation for a Tabular Form Row

To create a validation for a tabular form row:

 **Note:**

Text entered for validations may not exceed 3,950 characters.

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)."
2. To access the Create Validation Wizard. Under Page Processing, locate Validations. Click the **Create** icon.

The Create Validation Wizard appears.
3. For Validation Level:
 - a. Tabular Form - Select the associated tabular form region for this Validation.
 - b. Identify the validation level - Select **Tabular Form Row**.
 - c. Click **Next**.
4. For Sequence and Name specify the following and click **Next**.
 - a. Sequence - Enter the sequence number for this item.

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Under Validations, select the validation.
The Edit Page Validations page appears.
3. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help.
4. Click **Apply Changes**.

**See Also:**

["Viewing Field-Level Help](#) (page 1-27)"

15.2.4 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 in Page Designer](#) (page 15-10)"
- ["Editing a Validation in Component View](#) (page 15-18)"

15.2.5 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.

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 in Page Designer \(page 15-10\)"](#)

15.2.6 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 in Page Designer \(page 15-10\)"](#)
- ["Editing a Validation in Component View \(page 15-18\)"](#)
- ["Understanding Conditional Rendering and Processing \(page 2-10\)"](#)

15.2.7 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 \(page 20-35\)"](#)

15.2.8 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.

15.3 Understanding Page Processes

This section describes how to create a page process.

- [About Page Processes](#) (page 15-22)
- [Managing Page Processes in Page Designer](#) (page 15-22)
- [Managing Page Processes in Legacy Component View](#) (page 15-26)

15.3.1 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](#) (page 15-32)"
- ["Accessing Page Specific Utilities](#) (page 6-79)"

15.3.2 Managing Page Processes in Page Designer

This section describes how to create a page process.

See Also:

- ["Understanding Application Processes](#) (page 15-32)"
- ["Accessing Cross Page Utilities](#) (page 6-80)"

Tip:

To learn more about execution behavior in the case of a validation error, see ["What Happens When a Validation Fails?"](#) (page 15-21)."

- [Creating Page Processes in Page Designer](#) (page 15-23)
- [Editing Page Process Attributes in Page Designer](#) (page 15-24)
- [Controlling When a Page Process Executes in Page Designer](#) (page 15-25)
- [Creating Page Process Success and Error Messages in Page Designer](#) (page 15-25)
- [Making a Page Process Conditional in Page Designer](#) (page 15-26)

15.3.2.1 Creating Page Processes in Page Designer

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.

 **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 view all attributes in the Property Editor, select **Show All**.
4. To search for a group or an attribute, enter a keyword in the Filter Properties field.

 **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 the process.
 - b. Type - Specify which process type to execute. The Type you select determines what Setting attributes display.
6. Under Settings, edit the appropriate attributes.
7. Under Execution Options:
 - a. Sequence - Specify the sequence for this computation. The sequence determines the order of execution.

 **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**.

15.3.2.3 Controlling When a Page Process Executes in Page Designer

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 in Page Designer \(page 15-24\)"](#)

15.3.2.4 Creating Page Process Success and Error Messages in Page Designer

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 in Page Designer \(page 15-24\)"](#)
- ["Page Templates \(page 11-59\)"](#)

15.3.2.5 Making a Page Process Conditional in Page Designer

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 in Page Designer \(page 15-24\)"](#)

15.3.3 Managing Page Processes in Legacy Component View

Developers can create and edit page processes in Component View.

 **See Also:**

- ["Understanding Application Processes \(page 15-32\)"](#)

 **Tip:**

To learn more about execution behavior in the case of a validation error, see "[What Happens When a Validation Fails?](#) (page 15-21)"

- [Creating Page Processes in Component View](#) (page 15-27)
- [Editing Page Process Attributes in Component View](#) (page 15-29)
- [Controlling When a Page Process Executes](#) (page 15-30)
- [Creating Page Process Success and Error Messages](#) (page 15-30)
- [Making a Page Process Conditional](#) (page 15-31)

15.3.3.1 Creating Page Processes in Component View

This section describes how to create a page process. You create a process by running the Create Process Wizard. During the wizard, you define a process name, specify a sequence and the point at which the process executes, and select a process category. You can change nearly all of these attributes on the Edit Page Process page.

- [Creating a Page Process](#) (page 15-27)
- [Creating a Tabular Form Page Process](#) (page 15-28)

15.3.3.1.1 Creating a Page Process

To create a page process:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Create Page Process Wizard. Under Processes, click the **Create** icon.
3. For Process Type, select the category of the process you want to create as described in [Table 15-3](#) (page 15-27).

Table 15-3 Page Process Types

Process Category	Description
Clear Session State	Clears session state stored within Oracle Application Express. Select this process type to clear the cache for applications, sessions, or items and to clear existing user preferences. See Also: " Managing Session State Values (page 2-20)" and "Managing User Preferences" in <i>Oracle Application Express Administration Guide</i>
Close Dialog	Closes the current modal or non-modal dialog page.

Table 15-3 (Cont.) Page Process Types

Process Category	Description
Data Manipulation	Data Manipulation process types are frequently used by wizards to support Data Manipulation Language (DML) actions. App Builder supports the following declarative data manipulation processes: <ul style="list-style-type: none"> • Select Automatic Row Fetch and Automatic Row Processing (DML) to create an automatic Data Manipulation Language (DML) process. • Use Multi Row Update and Multi Row Delete with tabular forms. • Use Add Rows to Tabular Form with a tabular form.
Form Pagination	Process to retrieve the previous or next record from a database table or view. This process can be used to provide pagination on Form pages, so that end users can retrieve the previous or next record without navigating back to a report.
PL/SQL Code	Process to execute PL/SQL code you provide. Use this process type to execute a block of PL/SQL entered directly into the process or to simply call an existing API.
Plug-ins	Plug-ins allow developers to declaratively extend the built-in types available with Application Express. This option is only available when a plug-in type process exists in this application. See Also: " Implementing Plug-ins (page 18-3)."
Reset Pagination	Process to reset pagination of reports on the current page.
Send E-Mail	Process to send an email and optionally one or more attachments from the application.
User Preferences	Process to set User Preferences for the end user. See Also: "Managing User Preferences" in <i>Oracle Application Express Administration Guide</i>
Web Services	Implements a Web Service as a process on a page. Running the process submits a request to the service provider. See Also: " How to Invoke a Web Service as a Process (page 17-23)."

4. Follow the on-screen instructions.

15.3.3.1.2 Creating a Tabular Form Page Process

To create a tabular form page process:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Create Page Process Wizard. Under Processes, click the **Create** icon.
3. For Process Type:
 - Tabular Form - Select the associated tabular form region.
 - Process type - Select a process type as described in [Table 15-4](#) (page 15-29):

Table 15-4 Page Process Types

Process Category	Description
Clear Session State	Clears session state stored within Oracle Application Express. Select this process type to clear the cache for applications, sessions, or items and to clear existing user preferences. See Also: " Managing Session State Values (page 2-20)" and "Managing User Preferences" in <i>Oracle Application Express Administration Guide</i>
Close Dialog	Closes the current modal or non-modal dialog page.
Data Manipulation	Data Manipulation process types are frequently used by wizards to support Data Manipulation Language (DML) actions. App Builder supports the following declarative data manipulation processes: <ul style="list-style-type: none"> – Select Automatic Row Fetch and Automatic Row Processing (DML) to create an automatic Data Manipulation Language (DML) process. – Use Multi Row Update and Multi Row Delete with tabular forms. – Use Add Rows to Tabular Form with a tabular form.
Form Pagination	Process to retrieve the previous or next record from a database table or view. This process can be used to provide pagination on Form pages, so that end users can retrieve the previous or next record without navigating back to a report.
PL/SQL Code	Process to execute PL/SQL code you provide. Use this process type to execute a block of PL/SQL entered directly into the process or to simply call an existing API.
Plug-ins	Plug-ins allow developers to declaratively extend the built-in types available with Application Express. This option is only available when a plug-in type process exists in this application. See Also: " Implementing Plug-ins (page 18-3)."
Reset Pagination	Process to reset pagination of reports on the current page.
Send E-Mail	Process to send an email and optionally one or more attachments from the application.
User Preferences	Process to set User Preferences for the end user. See Also: "Managing User Preferences" in <i>Oracle Application Express Administration Guide</i>
Web Services	Implements a Web Service as a process on a page. Running the process submits a request to the service provider. See Also: " How to Invoke a Web Service as a Process (page 17-23)."

4. Follow the on-screen instructions.

15.3.3.2 Editing Page Process Attributes in Component View

Once you create a process, you can control when the process executes and what the process does by editing attributes on the Edit Page Process page.

To edit an existing page process:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View](#) (page 24-10)."
2. Access the Edit Page Process page. Under Processes, click the process name.
The Edit Page Process page appears.
3. Edit the appropriate attributes. For more information, see field-level Help and the sections that follow.
4. Click **Apply Changes**.

15.3.3.3 Controlling When a Page Process Executes

You control when a process executes by specifying a sequence number and a process point under Process Point on the Edit Page Process page. You can prevent a process from running during subsequent visits to a page by selecting the options under Run Process.

To control when a process executes:

1. Access the Edit Page Process page. See "[Editing Page Process Attributes in Component View](#) (page 15-29)."
2. Edit the following attributes:
 - a. Processing Point, Sequence - Specify the sequence for this component. The sequence determines the order of execution.
 - b. Processing Point, Process Point - Select the point in the processing at which this branch action is considered.
 - c. Processing Point, Run Process - Select how often this process executes.
3. Click **Apply Changes**.

15.3.3.4 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. Access the Edit Page Process page.
2. Edit the following attributes:
 - a. Messages, Process Success Message - Enter a message that displays when the process runs successfully.
 - b. Messages, Process Error Message - Enter the message that displays when an unhandled exception is raised. See field-level Help for more details of supported substitution strings for both success and error messages.
 - c. Messages, Error message display location - Identifies where the process error message displays. Process error messages can display on an error page, or inline with the existing page in the "notification" area (defined as part of the page template).

3. Click **Apply Changes**.

 **See Also:**

- ["Editing Page Process Attributes in Component View \(page 15-29\)"](#)
- ["Page Templates \(page 11-59\)"](#)

15.3.3.5 Making a Page Process Conditional

To create a conditional page process:

1. Access the Edit Page Process page.
2. Edit the following attributes:
 - a. Conditions, When Button Pressed - Select a page button. If a user clicks the button you select, the validation is performed (that is, depending upon to other defined conditions). This validation is performed if the page is submitted by a means other than the clicking the specified button.

If no button is pressed, other conditions are evaluated before performing the validation.
 - b. Conditions, 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.
 - c. Conditions, Condition Type - Select a condition type from the list that must be met in order for this component to be rendered processed.
 - Expression Attributes - Depending upon the selected Condition Type, enter values in the Expression attributes.
 - Execute Condition - Applicable only to tabular forms. Select either:
 - For Each Row** - Select this option if you refer to tabular form columns in your condition. This option causes the condition to evaluate each row of the tabular form.
 - Once** - Select this option if you do not refer to tabular form specific values.
3. Click **Apply Changes**.

 **See Also:**

- ["Editing Page Process Attributes in Component View \(page 15-29\)"](#)

15.4 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 \(page 14-25\)"](#)

15.5 Understanding Application Processes

This section describes how to create and manage application processes. Application processes are blocks of PL/SQL logic that are set to run at specific points using processes 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? \(page 15-21\)."](#)

- [On Demand Application Processes \(page 15-32\)](#)
- [About Application Processes that Execute On New Instance \(page 15-33\)](#)
- [About Running an On Demand Process from a Page Request \(page 15-33\)](#)
- [Application Process Example \(page 15-34\)](#)
- [Creating an Application Process \(page 15-34\)](#)
- [Editing Application Process Attributes \(page 15-35\)](#)
- [Creating Application Process Error Messages \(page 15-36\)](#)
- [Viewing the Application Processes History Report \(page 15-36\)](#)



See Also:

["Understanding Page Processes \(page 15-22\)"](#)

15.5.1 On Demand Application Processes

An **On Demand** process is special type of application process which has a Process Point of On Demand 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 in Component View \(page 15-27\)"](#)

15.5.2 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**).

15.5.3 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 \(page 2-22\)"](#)

15.5.4 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 \(page 12-3\)"](#)

15.5.5 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.

15.5.7 Creating Application Process Error Messages

If a process raises an error, you can define an error messages that displays to the user. How the 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 application processes, error messages display on a separate error page.

To create an application process error message:

1. Access the application process attributes.
2. Edit the following attributes:
 - a. **Source, Process Error Message** - Enter the message that displays when an unhandled exception is raised.
 - 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 \(page 15-35\)"](#)

15.5.8 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 \(page 8-94\)"](#)

15.6 Understanding Application Computations

This section describes how to create application computations.



Tip:

To learn more about execution behavior in the case of a validation error, see ["What Happens When a Validation Fails? \(page 15-21\)."](#)

- [About Application Computations \(page 15-37\)](#)
- [About Application Computations that Execute On New Instance \(page 15-38\)](#)
- [Creating an Application Computation \(page 15-38\)](#)
- [Accessing the Application Computation History Report \(page 15-39\)](#)

15.6.1 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;
```

15.6.2 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**).

15.6.3 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.

15.7 Using the Attribute Dictionary

This section explains how to use the Attribute Dictionary.

- [About the Attribute Dictionary](#) (page 15-40)
- [Accessing the Manage Attribute Dictionary](#) (page 15-40)

15.7.1 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.

You can 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.

15.7.2 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 both Page Designer and Component View, by clicking **Utilities** and then selecting **Attribute Dictionary**.

16

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.

Topics:

- [Working with Shared Components](#) (page 16-1)
- [Managing Component Settings](#) (page 16-9)
- [Managing Static Application Files](#) (page 16-12)
- [Managing Static Workspace Files](#) (page 16-17)
- [Managing Application-Level Items](#) (page 16-20)
- [Creating Lists of Values at the Application Level](#) (page 16-24)
- [Creating Lists](#) (page 16-27)
- [Using Shortcuts](#) (page 16-42)
- [Managing Report Output](#) (page 16-46)

See Also:

- ["App Builder Concepts](#) (page 2-1)"
- ["Creating Database Applications](#) (page 6-1)"
- ["Controlling Page Layout](#) (page 12-1)"
- ["Adding Navigation](#) (page 14-1)"

16.1 Working with Shared Components

Shared components are common elements that can display or be applied on any page within an application. You can use the tools and wizards on the Shared Components page either at the application-level or on specific pages.

- [About the Advantages of Using Shared Components](#) (page 16-2)
- [Accessing the Shared Components Page](#) (page 16-2)
- [About the Shared Components Icon](#) (page 16-2)
- [Shared Components Page](#) (page 16-3)
- [About the Advantages of Using Shared Components](#) (page 16-2)
- [Accessing the Shared Components Page](#) (page 16-2)

- [About the Shared Components Icon](#) (page 16-2)
- [Shared Components Page](#) (page 16-3)

**See Also:**

["Exporting an Application and Application Components](#) (page 23-16)"

16.1.1 About the Advantages of Using Shared Components

Tabs, lists, breadcrumbs, and navigation bar entries are shared components. Shared components are common elements that can display or be applied on any page within an application. You can use the tools and wizards on the Shared Components page either at the application-level or on specific pages. The big advantage of shared components is that once you create them, you can add them to any page within your database application.

**See Also:**

["Working with Shared Components](#) (page 16-1)"

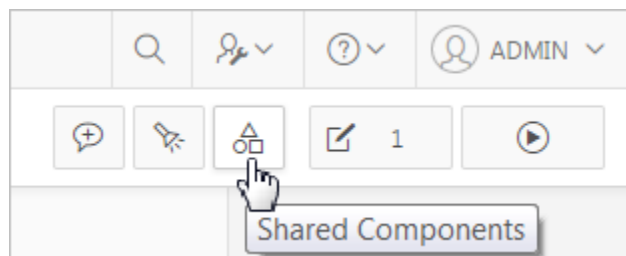
16.1.2 Accessing the Shared Components Page

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**.
The Shared Components page appears.
4. To create a shared component, select the appropriate link.

16.1.3 About the Shared Components Icon

Once you select an application, you 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, Component View, Supporting Objects, and Utilities.



16.1.4 Shared Components Page

Shared components are common elements that can display or be applied on any page within an application. Use the tools and wizards on the Shared Components page either at the application-level or on specific pages. The Shared Components is divided into the following regions.

- [Application Logic](#) (page 16-3)
- [Security](#) (page 16-4)
- [Other Components](#) (page 16-5)
- [Navigation](#) (page 16-5)
- [User Interface](#) (page 16-6)
- [Files](#) (page 16-7)
- [Data References](#) (page 16-7)
- [Reports](#) (page 16-7)
- [Globalization](#) (page 16-8)
- [Tasks Region](#) (page 16-8)



See Also:

["Accessing the Shared Components Page \(page 16-2\)"](#)

16.1.4.1 Application Logic

[Table 16-1](#) (page 16-3) describes the links under Application Logic on the Shared Components page.

Table 16-1 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 (page 6-16)"
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 (page 16-20)"

Table 16-1 (Cont.) Application Logic Links on the Shared Components Page

Link	Description
Application Processes	<p>Use application processes to run PL/SQL logic:</p> <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 <p>Note that On Demand processes execute only when called from a page-level On Demand process or when called using Ajax from the browser.</p> <p>See Also: "Understanding Application Processes (page 15-32)"</p>
Application Computations	<p>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.</p> <p>See Also: "Understanding Application Computations (page 15-37)"</p>

16.1.4.2 Security

[Table 16-2](#) (page 16-4) describes the links under Security on the Shared Components page.

Table 16-2 Security Links on the Shared Components Page

Link	Description
Security Attributes	<p>Use the Edit Security Attributes page to configure general security attributes for all pages within an application.</p> <p>See Also: "Configuring Security Attributes (page 6-23)"</p>
Authentication Schemes	<p>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.</p> <p>See Also: "Establishing User Identity Through Authentication (page 19-45)"</p>
Authorization Schemes	<p>Authorization restricts user access to specific controls or components based on predefined user privileges.</p> <p>See Also: "Providing Security Through Authorization (page 19-59)"</p>
Session State Protection	<p>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.</p> <p>See Also: "Understanding Session State Protection (page 19-26)"</p>

Table 16-2 (Cont.) Security Links on the Shared Components Page

Link	Description
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 (page 23-40)".

16.1.4.3 Other Components

[Table 16-3](#) (page 16-5) describes the links under Security on the Shared Components page.

Table 16-3 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 (page 16-24)"
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 (page 18-3)"
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 (page 16-9)"
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 (page 16-42)"

16.1.4.4 Navigation

[Table 16-4](#) (page 16-6) describes the links under Navigation on the Shared Components page.

Table 16-4 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 (page 16-27)"
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: " About Navigation Menus (page 16-28)"
Tabs	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 (page 14-2)"
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 (page 14-8)"
Navigation Bar Entries	Navigation bar entries 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. Navigation bar entries can display as a link from an image or text. A navigation bar entry can be an image, an image with text beneath it, or text. See Also: " Creating a Navigation Bar Entry (page 14-19)"

16.1.4.5 User Interface

[Table 16-5](#) (page 16-6) describes the links under User interface on the Shared Components page.

Table 16-5 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 (page 6-36)"
Themes	A theme is a named collection of templates that defines the application user interface. See Also: " Using Themes (page 11-1)"
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 (page 11-32)"

16.1.4.6 Files

[Table 16-6](#) (page 16-7) describes the links under User interface on the Shared Components page.

Table 16-6 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 (page 16-12)"
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 (page 16-17)"

16.1.4.7 Data References

[Table 16-7](#) (page 16-7) describes the links under User interface on the Shared Components page.

Table 16-7 Data References Links 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 (page 18-11)"
Web Service References	Web service references in App Builder are typically based on the Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) architectures. You can create a reference to a Web service and then incorporate it into an application to process data submitted by a form, or to render output in the form or report. See Also: " Implementing Web Services (page 17-8)"

16.1.4.8 Reports

[Table 16-8](#) (page 16-8) describes the links under User interface on the Shared Components page.

Table 16-8 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 (page 16-46)"
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 (page 16-49)"

16.1.4.9 Globalization

[Table 16-9](#) (page 16-8) describes the links under User interface on the Shared Components page.

Table 16-9 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 (page 6-32)" and " Understanding Application Translation and Globalization Support (page 20-1)"
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 Aso: " Translating Messages (page 20-15)"
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 (page 20-1)" and " Understanding the Translation Process (page 20-7)"

16.1.4.10 Tasks Region

The Tasks list on the right side of the Shared Components page contains these links:

- **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.

- **Manage Supporting Objects** - Links to the Supporting Objects page. Use this page to create a packaged application.
- **Edit Application Comments** - Links to the Application Comments page. Use this page to enter comments specific to the currently selected application.
- **Review Developer Comments** - Links to Developer Comments page.

 **See Also:**

- ["Exporting Application Components \(page 23-21\)"](#)
- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)
- ["Adding Database Application Comments \(page 6-71\)"](#)
- ["Adding Developer Comments \(page 6-68\)"](#)

16.2 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 \(page 16-9\)](#)
- [Integrating with Google Calendar \(page 16-10\)](#)
- [Configuring Color Picker Display \(page 16-10\)](#)
- [Configuring Interactive Report Action Menu Structure \(page 16-11\)](#)
- [Dynamically Adding Sliders to Regions \(page 16-11\)](#)
- [Configuring Switch Defaults \(page 16-11\)](#)
- [Viewing the Component Settings History Report \(page 16-12\)](#)

16.2.1 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.

16.2.2 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**.

16.2.3 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**.

16.2.4 Configuring Interactive Report Action Menu Structure

The structure of the interactive report Actions menu has changed to be consistent with interactive grids. Developers can configure Component Settings, Interactive Report to use the new structure or keep the legacy structure from earlier releases.

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**.

16.2.5 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**.

16.2.6 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 **Switch**.
5. 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**.

16.2.7 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.

16.3 Managing Static Application Files

This section describes 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](#) (page 16-13)
- [Uploading Static Application Files](#) (page 16-13)
- [Viewing the Static Application File Report](#) (page 16-14)
- [Referencing Static Application Files](#) (page 16-14)
- [Downloading Static Application Files](#) (page 16-15)
- [Deleting Static Application Files](#) (page 16-16)

 **See Also:**

- ["Managing Static Workspace Files](#) (page 16-17)"
- ["About Managing Images](#) (page 12-31)"

16.3.1 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://contentDeveliveryNetwork.com/myFiles/
```

5. Click **Apply Changes**.

16.3.2 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**.

16.3.3 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. To learn more, see "[About the Search Bar](#) (page 8-93)."

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**.

16.3.4 Referencing Static Application Files

Static application files are only available to the currently selected application. You can reference a static application file in your application using `#APP_IMAGES#` substitution string. 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 \(page 16-14\)"](#)
- ["About Using Substitution Strings \(page 2-26\)"](#)
- ["APP_IMAGES \(page 2-32\)"](#)
- ["Defining an Application Logo \(page 6-39\)"](#)

16.3.5 Downloading Static Application Files

This section describes how to download static application files.

- [Downloading a Static Application File \(page 16-15\)](#)
- [Downloading All Static Application Files in a Zip \(page 16-15\)](#)

16.3.5.1 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.

16.3.5.2 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**.

16.3.6 Deleting Static Application Files

This section describes how to delete uploaded static application files.

- [Deleting a Static Application File](#) (page 16-16)
- [Deleting All Static Application Files](#) (page 16-16)

16.3.6.1 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 \(page 8-93\)"](#)

16.3.6.2 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**.

16.4 Managing Static Workspace Files

This section describes 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](#) (page 16-17)
- [Viewing Static Workspace File Report](#) (page 16-18)
- [About Referencing Static Workspace Files](#) (page 16-18)
- [Downloading Static Workspace Files](#) (page 16-18)
- [Deleting Static Workspace Files](#) (page 16-19)

See Also:

- ["Managing Static Application Files](#) (page 16-12)"
- ["About Managing Images](#) (page 12-31)"

16.4.1 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**.

16.4.2 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 \(page 8-93\)"](#)

16.4.3 About Referencing Static Workspace Files

You can reference static workspace files in any application in the current workspace. You can reference a static workspace file in your application using `#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 \(page 16-18\)"](#)
- ["Using Built-in Substitution Strings \(page 2-28\)"](#)
- ["WORKSPACE_IMAGES \(page 2-46\)"](#)
- ["Defining an Application Logo \(page 6-39\)"](#)

16.4.4 Downloading Static Workspace Files

This section describes how to download static workspace files.

- [Downloading a Static Workspace File](#) (page 16-19)
- [Downloading All Static Workspace Files in a Zip](#) (page 16-19)

16.4.4.1 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.

16.4.4.2 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**.

16.4.5 Deleting Static Workspace Files

This section describes how to delete uploaded static workspace files.

- [Deleting a Static Workspace File](#) (page 16-19)
- [Deleting All Static Workspace Files](#) (page 16-20)

16.4.5.1 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 \(page 8-93\)"](#)

16.4.5.2 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**.

16.5 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 \(page 16-21\)](#)
- [Viewing Application Item Usage \(page 16-22\)](#)
- [Editing Application-Level Item Attributes \(page 16-23\)](#)
- [About Creating an Application-Level Item that Functions as a Global Variable \(page 16-23\)](#)

 **See Also:**

- ["About the Differences Between Page Items and Application Items \(page 13-2\)"](#)
- ["Referencing Item Values \(page 13-2\)"](#)

16.5.1 Creating an Application-Level Item

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 \(page 16-23\)"](#)

16.5.2 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.

16.5.3 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 \(page 1-27\)"](#)
- ["Understanding Cross-Site Scripting Protection \(page 19-21\)"](#)

16.5.4 About Creating an Application-Level Item that Functions as a Global Variable

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.

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.

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 \(page 16-21\)"](#)

16.6 Creating Lists of Values at the Application Level

This section describes how to create lists of values.

- [What is a List of Values? \(page 16-24\)](#)
- [Creating a Named LOV at the Application Level \(page 16-25\)](#)
- [About Static LOVs \(page 16-25\)](#)
- [Editing an Existing LOV \(page 16-25\)](#)
- [About Referencing Session State Within an LOV \(page 16-26\)](#)
- [Accessing LOV Reports \(page 16-26\)](#)

16.6.1 What is a List of Values?

A list of values (LOV) is a static or dynamic set of values used to display a specific type of page item, 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 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.



See Also:

["Creating Page-Level Items \(page 13-9\)"](#) and ["Working with Shared Components \(page 16-1\)"](#)

16.6.2 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.

16.6.3 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.

16.6.4 Editing an Existing LOV

To edit an existing LOV, select the LOV on the Lists of Values page.

- [Editing a LOV](#) (page 16-25)
- [Editing LOVs in Bulk](#) (page 16-26)

16.6.4.1 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.
To learn more about an attribute, see field-level Help.
5. Click **Apply Changes**.



See Also:

["Viewing Field-Level Help \(page 1-27\)"](#)

16.6.4.2 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**.

16.6.5 About Referencing Session State Within an LOV

You can reference session state 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
```

16.6.6 Accessing LOV Reports

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:
 - **Edit Null Text** - Edit null text for all LOVs in your application.
 - **Locally Defined** - 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.
 - **Grid Edit** - Enables you to edit the display values of all static LOVs.
 - **Conditional Entries** - Displays all LOVs that display conditionally.
 - **Subscription** - Displays all subscribed LOVs in your application.
 - **Utilization** - Displays where LOVs are used in your application.
 - **History** - Displays a history of recently changed LOVs by date.
4. Follow the on-screen instructions.

16.7 Creating Lists

Developers can add shared collection of links (or a list) to a database application. 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.

- [About Creating Lists](#) (page 16-27)
- [About Navigation Menus](#) (page 16-28)
- [Creating Static Lists](#) (page 16-29)
- [Creating Dynamic Lists](#) (page 16-35)
- [Copying a List](#) (page 16-38)
- [Adding a List to a Page](#) (page 16-39)
- [Editing List Attributes](#) (page 16-40)
- [Accessing List Reports](#) (page 16-41)

 **See Also:**

s

16.7.1 About Creating Lists

Oracle Application Express supports two types of lists:

- **Static Lists** - 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.

- **Dynamic Lists** - Dynamic lists are 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 such as jQuery Mobile.

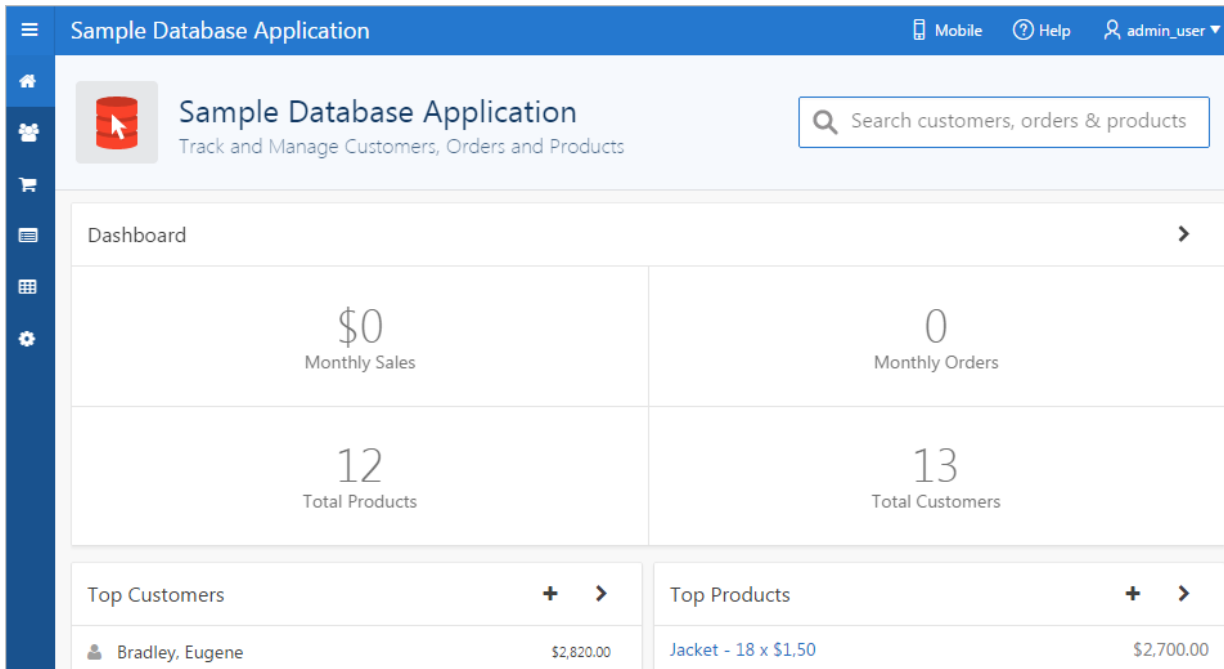
The list definition displays a specific type of page item, such as progress bars, sidebar, bullet navigation list, or navigation menu. You can control how a list displays through templates.



See Also:

- ["Creating Static Lists \(page 16-29\)"](#)
- ["Creating Dynamic Lists \(page 16-35\)"](#)
- ["Understanding Dynamic List Syntax \(page 16-36\)"](#)
- ["Creating a New Template \(page 11-35\)"](#)
- ["List Templates \(page 11-56\)"](#)

16.7.2 About Navigation Menus

Older themes provide navigation support through the creation of tabs. Newer themes, such as *Universal Theme - 42*, provide navigation support through the creation of navigation menus. If the theme you are using supports navigation menus, you can create navigation menus automatically when running the Create Application Wizard. The Sample Database Application includes a navigation menu that displays in a sidebar.



Sample Database Application		Mobile	Help	admin_user
	Sample Database Application Track and Manage Customers, Orders and Products	<input type="text" value="Search customers, orders & products"/>		
Dashboard		>		
\$0 Monthly Sales	0 Monthly Orders			
12 Total Products	13 Total Customers			
Top Customers	+ >	Top Products	+ >	
 Bradley, Eugene	\$2,820.00	Jacket - 18 x \$1.50	\$2,700.00	

Navigation menus support the creation of hierarchical sub-menus. You can render a navigation menu either at the top of the page as tabs or as a side bar (as shown in the previous example). Navigation menus displaying as a side bar are responsive. Based on the available space, the navigation bar displays a full menu or collapses to a narrow icon bar. Side navigation menus are controlled by the theme and user interface level.

You can select an alternative list template or change the position of a navigation menu by editing User Interface Details. You can configure the available navigation menu list templates by editing the theme.

About Switching from Tabs to Navigation Menus

Developers 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:

- ["Managing the Application User Interface \(page 6-36\)"](#)
- ["Editing Themes \(page 11-10\)"](#)

16.7.3 Creating Static Lists

This section explains about to create a static list.

- [About the Process of Creating a Static List \(page 16-29\)](#)
- [Creating a Static List from Scratch \(page 16-30\)](#)
- [About Adding Entries and Sublists to a Static List \(page 16-31\)](#)
- [Adding Entries or Sublists from Scratch \(page 16-31\)](#)
- [Copying Static List Entries Between Lists \(page 16-33\)](#)
- [Reparenting Static List Entries \(page 16-34\)](#)
- [Managing Orphaned Static List Entries \(page 16-34\)](#)
- [Resequencing Static List Entries \(page 16-35\)](#)

16.7.3.1 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 \(page 16-39\)"](#)
- ["Editing List Attributes \(page 16-40\)"](#)
- ["Accessing List Reports \(page 16-41\)"](#)

16.7.3.2 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**.

16.7.3.3 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 \(page 11-32\)"](#)
- ["List Templates \(page 11-56\)"](#)

16.7.3.4 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 \(page 11-10\)"](#)
- ["Optimizing a Page for Printing \(page 12-2\)"](#)
- ["COUNT_CLICK Procedure" in *Oracle Application Express API Reference*](#)

16.7.3.5 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**.

16.7.3.6 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**.

16.7.3.7 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**.

16.7.3.8 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**.

16.7.4 Creating Dynamic Lists

This section explains about to create a dynamic list.

- [About the Process of Creating a Dynamic List](#) (page 16-35)
- [Understanding Dynamic List Syntax](#) (page 16-36)
- [Creating a Dynamic List from Scratch](#) (page 16-37)

16.7.4.1 About the Process of Creating a Dynamic List

A dynamic list enables you to dynamically create styled list items that support mobile frameworks such as jQuery Mobile. Dynamic lists are based on a SQL query or a PL/SQL function returning a SQL query.

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 \(page 16-39\)"](#)
- ["Editing List Attributes \(page 16-40\)"](#)
- ["Accessing List Reports \(page 16-41\)"](#)

16.7.4.2 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.

16.7.4.3 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 \(page 16-36\)"](#)

16.7.5 Copying a List

This section discusses how to copy 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 \(page 16-38\)](#)
- [Copying a List Between Applications \(page 16-39\)](#)

16.7.5.1 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**.

16.7.5.2 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**.

16.7.6 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 region and specifying the region type as List.



See Also:

["Creating a Region in Page Designer \(page 12-5\)"](#)

To add a list to a page:

1. Navigate to the appropriate page. See "[Viewing a Page in Legacy Component View \(page 24-10\)](#)".
2. Under Regions, click the **Create** icon.
The Create Region Wizard appears.
3. For Region, select **List** as the region type and click **Next**.
4. For Display Attributes:
 - a. Title - Enter a title for the region. This title displays if the region template you choose displays the region title.

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**.

16.7.8 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.

16.8 Using Shortcuts

This section describes how to create and use shortcuts.

- [What is a Shortcut?](#) (page 16-42)
- [Understanding Shortcut Types](#) (page 16-42)
- [Defining and Referencing Shortcuts](#) (page 16-43)
- [Editing Existing Shortcuts](#) (page 16-45)
- [Accessing Shortcut Reports](#) (page 16-45)

16.8.1 What is a Shortcut?

Using shortcuts enables you to avoid repetitive coding of HTML or PL/SQL functions. You can use a shortcut to define a page control such as a button, HTML text, a PL/SQL procedure, or HTML. Once defined, you can invoke a shortcut using specific syntax unique to the location in which the shortcut is used. Shortcuts can be referenced many times, thus reducing code redundancy.

You can use shortcut within the following locations:

- The Region Source attribute of regions defined as HTML Text (with shortcuts). See "[About Regions](#) (page 12-4)."
- Region Header and Footer Text attribute. See "[Specifying a Region Header and Footer](#) (page 12-8)."
- Item Label attributes, Pre Element Text, Post Element Text, and Default Value attribute.
- Region Templates attributes. See "[Creating Custom Themes](#) (page 11-32)."

16.8.2 Understanding Shortcut Types

This section describes the types of shortcut supported by Oracle Application Express.

- [Supported Shortcut Types](#) (page 16-42)
- [Text with JavaScript Escaped Single Quotes](#) (page 16-43)
- [Message](#) (page 16-43)
- [Message with JavaScript Escaped Single Quotes](#) (page 16-43)

16.8.2.1 Supported Shortcut Types

When you create a shortcut you select a shortcut type. Oracle Application Express supports the following types of shortcuts:

- 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

16.8.2.2 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.

16.8.2.3 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.

16.8.2.4 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 \(page 20-1\)"](#)

16.8.3 Defining and Referencing Shortcuts

Before you can incorporate a shortcut in your application, you must define it and add it to the Shortcuts repository. This section describes how to define and reference a shortcut.

- [Defining a Shortcut from Scratch](#) (page 16-43)
- [Copying an Existing Shortcuts](#) (page 16-44)
- [Referencing a Shortcut](#) (page 16-45)

16.8.3.1 Defining a Shortcut from Scratch

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).

16.8.3.2 Copying an Existing Shortcuts

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).

16.8.3.3 Referencing a Shortcut

You reference shortcuts using the following syntax:

```
"MY_SHORTCUT"
```

The shortcut name must be capitalized and enclosed in quotation marks.

16.8.4 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 \(page 1-27\)"](#)

16.8.5 Accessing Shortcut 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.

16.9 Managing Report Output

This section explains 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](#) (page 16-46)
- [Formatting a Report Region or Report Query Using Report Layouts](#) (page 16-49)

16.9.1 Printing a Report Region by Defining a Report Query

This section explains how to print a report region by defining a report query.

- [About Report Queries](#) (page 16-46)
- [Creating a Report Query](#) (page 16-46)
- [Editing a Report Query](#) (page 16-49)
- [Copying a Report Query](#) (page 16-49)

16.9.1.1 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.

16.9.1.2 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.



See Also:

["About Report Printing Configuration Options \(page 8-124\)"](#)

16.9.1.3 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**.

16.9.1.4 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.

16.9.2 Formatting a Report Region or Report Query Using Report Layouts

This section describes how to use a report layout to format a report region or report layout.

- [About Report Layouts](#) (page 16-50)
- [About Report Layout Options](#) (page 16-50)
- [Creating a Report Layout](#) (page 16-51)

- [Editing a Report Layout](#) (page 16-52)
- [Copying a Report Layout](#) (page 16-52)

16.9.2.1 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*

16.9.2.2 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*.

16.9.2.3 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.

17

Managing Application Data

This section provides information on managing application data including creating applications with data loading capability, using collections, implementing Web services, accessing data with database links, and utilizing Data Manipulation Language (DML) to update or delete rows of a table.

Topics:

- [Creating Applications with Data Loading Capability](#) (page 17-1)
- [About Using Collections](#) (page 17-7)
- [Implementing Web Services](#) (page 17-8)
- [Accessing Data with Database Links](#) (page 17-30)
- [About DML Locking](#) (page 17-32)

17.1 Creating Applications with Data Loading Capability

Developers can 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](#) (page 17-1)
- [Creating a Data Load Wizard](#) (page 17-3)
- [Re-creating a Data Load Pages](#) (page 17-5)
- [Editing the Data Load Definition](#) (page 17-5)

See Also:

"Data Loading Wizard Examples" in *Oracle Application Express End User Guide* to learn about using the Data Load Wizard.

17.1.1 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 creation process, 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 \(page 17-3\)"](#)
- ["Editing the Data Load Definition \(page 17-5\)"](#)

17.1.2 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.
 - b. Select a page type - Select **Data Loading**.
5. For Data Load Table:
 - a. Definition Name - Enter the name of this data load definition.
 - b. Owner - Select the owner of the data loading table.
 - c. Table Name - Select the table to use for data loading, also referred to as the upload table.
 - d. Unique Column 1 - Select the column name to use as the primary unique key column during the data load process. At least one unique key column is required.
 - e. Case Sensitive - Identify whether the selected unique key column is case sensitive. By default, this is set to **No**.
 - f. Unique Column 2 - If the unique key definition of the data load table is a compound key consisting of two or more columns, select the column name to use as the second unique key column during the data load process.
 - g. Case Sensitive - Identify whether the selected unique key column is case sensitive. By default, this is set to **No**.

 **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 a Data Load Pages \(page 17-5\)"](#)
- ["Managing Pages in Page Designer \(page 7-1\)"](#)

17.1.3 Re-creating a 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**.

17.1.4 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**.

- 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**.

17.2 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*

17.3 Implementing Web Services

This section describes how to utilize Web services in Oracle Application Express applications.

 **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](#) (page 17-8)
- [Creating Web Service References in Oracle Application Express](#) (page 17-9)
- [About Working with SSL Enabled Web Services](#) (page 17-10)
- [Creating Web Service References Based on a WSDL](#) (page 17-10)
- [Creating Web Service References Manually](#) (page 17-12)
- [Creating RESTful Web Service References](#) (page 17-14)
- [Using the Web Service Reference Repository](#) (page 17-17)
- [Creating an Input Form and Report on a Web Service](#) (page 17-17)
- [Creating a Form on a Web Service](#) (page 17-20)
- [How to Invoke a Web Service as a Process](#) (page 17-23)
- [Editing a Web Service Process](#) (page 17-26)
- [Viewing a Web Service Reference History](#) (page 17-27)
- [Exposing a Report Region as a RESTful Web Service](#) (page 17-27)

17.3.1 About Web Services

Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment. In a typical Web services scenario, a business application sends a request to a service at a given URL by using the protocol over HTTP. The service receives the request, processes it, and returns a response. You can incorporate calls with external Web services in applications developed in 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 \(page 19-14\)"](#)

17.3.2 Creating Web Service References in Oracle Application Express

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 \(page 17-9\)](#)
- [Specifying an Application Proxy Server Address \(page 17-10\)](#)

 **See Also:**

- ["Creating Web Service References Based on a WSDL \(page 17-10\)"](#)
- ["Creating Web Service References Manually \(page 17-12\)"](#)
- ["Creating RESTful Web Service References \(page 17-14\)"](#)

17.3.2.1 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 References, click **Web Service References**.
The Web Service References page appears.

17.3.2.2 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**.

17.3.3 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*

17.3.4 Creating Web Service References Based on a WSDL

This section describes how to create Web service reference based on a WSDL.

- [About Creating Web Service References Based on a WSDL](#) (page 17-11)

- [Creating a Web Service Reference by Specifying a WSDL Document](#) (page 17-11)
- [Testing a Web Service Reference Created from a WSDL](#) (page 17-12)

17.3.4.1 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.

When you create a Web service reference based on a WSDL, the wizard 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

17.3.4.2 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.
2. Click **Create**.
3. For Web Reference Type, select **Based on WSDL** and click **Next**.
4. When 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.

 **See Also:**

- ["Accessing the Web Service References Page \(page 17-9\)"](#)
- ["Creating a Form After Creating a Reference \(page 17-21\)"](#)
- ["Creating a Form and Report After Creating a Reference \(page 17-18\)"](#)

17.3.4.3 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.
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 \(page 17-9\)"](#)

17.3.5 Creating Web Service References Manually

This section describes how to create a Web service reference manually.

- [About Creating Web Service References Manually \(page 17-13\)](#)
- [Creating a Web Service Reference Manually \(page 17-13\)](#)
- [Testing a Web Service Reference Created Manually \(page 17-14\)](#)

17.3.5.1 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

17.3.5.2 Creating a Web Service Reference Manually

To create a Web service reference manually:

The Web service reference is added to the Web Service References Repository.

1. Navigate to the Web Service References page.
2. Click **Create**.
3. Select **Manual** and click **Next**.

The Create/Edit Web Service page appears.

4. Under Web Service, for Name, enter a name to identify the reference.
5. Under Service Description, make these changes:
 - a. 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.
 - b. 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.
 - c. Proxy - Enter a proxy to override the application proxy for this service.
 - d. SOAP Version - Select **1.1** or **1.2**.
 - e. Basic Authentication - Choose whether the Web service requires authentication. Select **Yes** or **No**.
6. Under SOAP Envelope, for SOAP Envelope, enter 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.

7. Under SOAP Response, for Store Response in Collection, enter the name of a collection to store the response (optional). The response is stored in the `CLOB001` column.
8. Click **Create**.



See Also:

["Accessing the Web Service References Page \(page 17-9\)"](#)

17.3.5.3 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.
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 \(page 17-9\)"](#)

17.3.6 Creating RESTful Web Service References

This section describes how to create a RESTful Web service reference.

- [About Creating RESTful Web Service References \(page 17-14\)](#)
- [Creating a RESTful Web Service Reference \(page 17-15\)](#)
- [Testing a REST Web Service Reference \(page 17-16\)](#)

17.3.6.1 About Creating RESTful Web Service References

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

17.3.6.2 Creating a RESTful Web Service Reference

To create a RESTful Web service reference:

1. Navigate to the Web Service References page.
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 \(page 17-9\)"](#)
- ["Creating a Proxy Server for an Instance" in *Oracle Application Express Administration Guide*](#)

17.3.6.3 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.
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 \(page 17-9\)"](#)

17.3.7 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.

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 \(page 17-9\)"](#)
- ["About the Actions Menu \(page 8-94\)"](#)

17.3.8 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 \(page 17-18\)](#)
- [Creating a Form and Report After Creating a Reference \(page 17-18\)](#)
- [Creating a Form and Report by Adding a New Page \(page 17-19\)](#)

17.3.8.1 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.

17.3.8.2 Creating a Form and Report After Creating a Reference

To create a form and report after creating a Web service reference:

1. Create the Web service reference.
2. After the Web service reference has been added, select **Create Form & Report on Web Service**.
3. For Choose Service and Operation, make these selections and click **Next**:
 - a. Web Service Reference - Select the Web service reference.
 - b. Operation - Select the method to be executed. For REST style Web references, select **doREST**.
 - c. User Interface - Select Desktop to create a form and report for a desktop. Select jQuery Mobile Smartphone to create a form and report for a mobile phone. The jQuery Mobile Smartphone option is only available if the user interface options for the application properties have jQuery Mobile Smartphone selected.
4. For Page and Region Attributes, review the displayed attributes and click **Next**.

**Note:**

If the page you specify does not exist, the wizard creates the page for you.

5. 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.
6. 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.

7. For Window Service Results, make these selections and click **Next**:
 - a. Temporary Result Set Name - Enter a name for the collection that stores the Web service result.
 - b. Result Tree to Report On (Not displayed for REST style Web service references) - Select the portion of the resulting XML document that contains the information you want to include in the report.
8. For Report Parameters, select the parameters to be included in the report and click **Next**.
9. For Tab Options, select whether the page should have tabs and how they should be created, then click **Next**.
10. Review confirmation page and click **Create**.

 **See Also:**

- ["Creating Web Service References Based on a WSDL \(page 17-10\)"](#)
- ["Creating RESTful Web Service References \(page 17-14\)"](#)

17.3.8.3 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. Create a new page.

In the Create Page Wizard:

- a. For User Interface, select Desktop to create a form and report for a desktop. Select jQuery Mobile Smartphone to create a form and report for a mobile phone. The jQuery Mobile Smartphone option is only available if the user interface options for the application properties have jQuery Mobile Smartphone selected.
- b. Select **Form** and click **Next**.
- c. Select **Form and Report on Web Service** and click **Next**.
3. For Choose Service and Operation, make the following selections and click **Next**:
 - 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.
4. For Page and Region Attributes, review 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, make the following 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.
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 **Finish**.

 **See Also:**

- ["Creating Web Service References Based on a WSDL \(page 17-10\)"](#)
- ["Creating Web Service References Manually \(page 17-12\)"](#)
- ["Creating RESTful Web Service References \(page 17-14\)"](#)

17.3.9 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 \(page 17-20\)](#)
- [Creating a Form After Creating a Reference \(page 17-21\)](#)
- [Creating a Form by Adding a New Page \(page 17-22\)](#)

17.3.9.1 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.

17.3.9.2 Creating a Form After Creating a Reference

To create a form after creating a Web service reference:

1. Create the Web service reference.
2. After the Web service reference has been added, select **Create Form on Web Service**.
3. For Choose Service and Operation, Make the following selections then click **Next**:
 - 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. User Interface - Select Desktop to create a form and report for a desktop. Select jQuery Mobile Smartphone to create a form and report for a mobile phone. The jQuery Mobile Smartphone option is only available if the user interface options for the application properties have jQuery Mobile Smartphone selected.
4. For Page and Region Attributes, review 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, 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.
6. 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.

7. 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.
8. 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**.

- 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.
6. 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.

7. 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.
8. 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.
9. For Tab Options, select whether the page should have tabs and how they should be created, then click **Next**.
10. Review confirmation page and click **Create**.

 **See Also:**

- ["Creating Web Service References Based on a WSDL \(page 17-10\)"](#)
- ["Creating Web Service References Manually \(page 17-12\)"](#)
- ["Creating Web Service References Manually \(page 17-12\)"](#)
- ["Managing Pages in a Database Application \(page 6-44\)"](#)
- ["Managing the Application User Interface \(page 6-36\)"](#)

17.3.10 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 \(page 17-24\)](#)
- [Displaying Web Service Results in a Report \(page 17-25\)](#)

17.3.10.1 Invoking a Web Service as a Process

To invoke a Web service as a process:

1. Create a Web Service Reference. See "[Creating Web Service References Based on a WSDL](#) (page 17-10)," "[Creating Web Service References Manually](#) (page 17-12)," or "[Creating a RESTful Web Service Reference](#) (page 17-15)."

2. Create a page.

In the Create Page Wizard:

- a. Select **Blank Page**.
 - b. If prompted to use a navigation menu, select **No**.
3. Click **Edit Page**.
 4. Under Page Rendering or Page Processing, locate the Processes section, right click and select **Create Process**.

The Create Page Process wizard appears.

5. For Process Type, select **Web Services** and click **Next**.
6. For Process Attributes, specify the following and click **Next**:

- a. Name - Enter the process name.
- b. Sequence - Enter the sequence for this component. This specifies the order of evaluation.
- c. Point - Select the processing point.

7. For Process, specify the following:

- a. Web Service Reference - Select the Web service reference for this process.
If the Web reference was created from a WSDL or is RESTful style, the Operation selection appears.
- b. Operation (WSDL or RESTful style Web service references only) - Select the Web service operation this process executes.

The Web Service Input Parameters and Web Service Output Parameters appear.

- c. Under Web Service Input Parameters, specify inputs to the Web service process:

- Source - Select the source type.
- Value - Select the source value.

- d. Under Web Service Output Parameters, specify where the results are stored:

To store the results in a collection:

- Store Result in - Select **Collection**.
- Value - Enter a name for the collection.

To store the results in items on the page:

- Store Result in - Select **Items**.
- Value - Enter the appropriate item value.

- e. Click **Next**.
8. For Messages, make the following entries and click **Next**:
 - a. Success Message - Enter the message that displays on the next page when the process runs successfully.
 - b. Failure Message - Enter the message that displays when an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays.
9. For Process Conditions, make the following selections and click **Next**:
 - a. When Button Pressed - Select that button that causes the process to run.
 - b. Condition Type - Select a condition type that must be met in order for this component to be processed. If the condition type requires expressions for comparison, the Expression 1 and possibly Expression 2 fields display. Enter expressions into these fields as required.
10. Click **Create Process**.

 **See Also:**

- ["Managing Pages in a Database Application \(page 6-44\)"](#)
- ["Accessing Page Attributes in Legacy Component View \(page 24-12\)"](#)

17.3.10.2 Displaying Web Service Results in a Report

To create a report in which to display Web Service request results:

1. Navigate to the page.
2. In component view, for Regions, click the **Create** icon.
The Create Region Wizard appears.
3. For the region type, select **Report** and click **Next**.
4. For Region, select **Web Service Result** and click **Next**.
5. For Display Attributes, enter a region title and optionally edit the region attributes and click **Next**.
6. Choose whether the Web reference was created manually, from a WSDL, or is RESTful style and click **Next**.
7. If the Web service reference was created from a WSDL:
 - a. For Web Service Reference, select the Web service reference to execute.
 - b. For Operation, select the method to be executed.
 - c. For Result Tree to Report On, select the portion of the resulting XML document that contains the information you want to include in the report.
 - d. For Report Parameters and Options:
 - In Temporary Result Set Name, enter a name for the collection that stores the Web service result.
 - Select and deselect the appropriate parameters.

8. If the Web service reference was created manually:
 - a. Select the Web service reference.
 - b. Choose the SOAP style.
 - c. Choose the message format.
 - d. Enter the XPath expression to the node to report on.
 - e. Enter the namespace for the SOAP response envelope and click **Next**.
 - f. Enter the name of the collection where the response message is stored.
 - g. Enter the names of the parameters that you want to be included in the report.
9. If the Web service reference is REST style:
 - a. Select the Web service reference.
 - b. Enter the name of the collection where the response message is stored.
 - c. Choose the report template and the number of rows to display per page.
 - d. Select the parameters to be included in the report.
10. Click **Create SQL Report**.

**See Also:**

["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

17.3.11 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.
3. Select the process name.

The Edit Page Process page appears.
4. To map an input parameter to a static value:
 - a. Scroll down to Web Service Input Parameters.
 - b. Enter a value in the Value field, adjacent to the appropriate parameter name.
5. To map an output parameter to a collection:
 - a. Scroll down to Web Service Output Parameters.
 - b. For Store Response In, select **Collection**.
 - c. For Value, enter the name of a collection to store the response. The response is stored in the `CLOB001` or `XMLTYPE001` based on the expected response.
6. To map an output parameter to an item:

- a. Scroll down to Web Service Output Parameters.
 - b. For Store Response In, select **Items**.
 - c. Enter an item name in the Value field, adjacent to the appropriate parameter name.
7. Click **Apply Changes**.

 **See Also:**

["How to Invoke a Web Service as a Process \(page 17-23\)"](#)

17.3.12 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. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
The Shared Components page appears.
4. Under Logic, click **Web Service References**.
5. Click **History**.

 **Note:**

The History button only appears on the Web Service Reference page if at least one reference exists.

17.3.13 Exposing a Report Region as a RESTful Web Service

This section describes how to expose a report region as a RESTful Web service.

- [About Exposing a Report Region as a RESTful Web Service \(page 17-28\)](#)
- [Enabling RESTful Access to a Report Region \(page 17-28\)](#)
- [About Accessing a RESTful Enabled Report Region from a Web Service Client \(page 17-29\)](#)

 **See Also:**

"Using RESTful Services" in *Oracle Application Express SQL Workshop Guide*

17.3.13.1 About Exposing a Report Region as a RESTful Web Service

To expose a report region as a RESTful Web service, perform the following steps:

- Step 1: The Instance administrator must enable the REST provider feature in instance settings.
- Step 2: A developer must enable RESTful access to a report region.
- Step 3: The page that contains the report must not require authentication.

See Also:

- "Enabling RESTful Access" in the *Oracle Application Express Administration Guide*
- "[Enabling RESTful Access to a Report Region](#) (page 17-28)"

17.3.13.2 Enabling RESTful Access to a Report Region

To enable RESTful access to a report region, you must make the page public so no authentication is required and enable the report for RESTful access.

Tip:

To enable RESTful access to an Oracle Application Express instance, see "Enabling RESTful Access" in the *Oracle Application Express Administration Guide*.

To enable RESTful Access to a report region:

1. On the Workspace home page, click **App Builder**.
The App Builder home page appears.
2. Select an application.
3. View the page in Component View that contains the report you want to enable.
4. Make the page public:
 - a. Under Page, click the **Edit All** icon.
 - b. Under Security, select **Page is Public** from the Authentication list.
 - c. Click **Apply Changes**.
5. Enable a report region for RESTful access:
 - a. Under Regions, click the name of the region that contains the report you want to enable.
 - b. Under Attributes, enter a value for Static ID field. This value is used to access the report RESTfully.
 - c. From the Enable RESTful Access List, select **Yes**.

 **Note:**

The Enable RESTful Access List option only displays if RESTful access to this Oracle Application Express instance has been enabled.

d. Click **Apply Changes**. **See Also:**

- "[Viewing a Page in Legacy Component View](#) (page 24-10)"
- "Enabling RESTful Access" in the *Oracle Application Express Administration Guide*

17.3.13.3 About Accessing a RESTful Enabled Report Region from a Web Service Client

Once you enable a report for RESTful access, you need to know the endpoint URL and parameters to pass to the RESTful Web service. The endpoint URL is similar to the URL used to access this instance of Oracle Application Express followed by the resource `apex_rest.getReport`. For example:

```
http://apex.oracle.com/apex/apex_rest.getReport
```

 **Tip:**

If your client uses the POST method, you must also set the HTTP Header, `Content-Type`, to the value `application/x-www-form-urlencoded`.

Name	Default	Required	Description
<code>app</code>	N/A	Yes	The numeric ID or alias of the application that contains the RESTful enabled report.
<code>page</code>	N/A	Yes	The numeric ID or alias of the page that contains the RESTful enabled report.
<code>reportid</code>	N/A	Yes	The Static ID attribute of the RESTful enabled report.
<code>parmvalues</code>	null	No	Values for the report parameters can be sent in a comma separated list. For example: <code>CLERK,10</code>
<code>lang</code>	en	No	Sets the NLS environment variables before running the report based on the language passed in. For example: <code>de</code>

Name	Default	Required	Description
output	xml	Yes	This determines whether XML or JSON are returned to the client. Domain of possible values are: xml, json

The `apex_rest` service also has an operation to allow discoverability of RESTful enabled reports, given an application ID or alias. The response is an XML document with a description of all reports that can be accessed by RESTful Web services. This service is invoked with a URL similar to the following:

```
http://apex.oracle.com/apex/apex_rest.getServiceDescription?app=691
```

In the URL above, 691 is the numeric ID of an application. The document returned is similar to the following:

```
<?xml version="1.0"?>
<urn:getServiceDescriptionResponse xmlns:urn="urn:oasis:names:tc:wsrp:v1:types">
  <urn:requiresRegistration>false</urn:requiresRegistration>
  <urn:offeredPortlets>
    <urn:PortletDescription>
      <urn:portletHandle>employees</urn:portletHandle>
      <urn:markupTypes>
        <urn:mimeType>application/xml</urn:mimeType>
        <urn:mimeType>application/json</urn:mimeType>
      </urn:markupTypes>
      <urn:groupID>1</urn:groupID>
      <urn:description/>
      <urn:title>EMP</urn:title>
      <urn:keywords>
        <urn:value>PL_JOB</urn:value>
        <urn:value>PL_DEPTNO</urn:value>
      </urn:keywords>
    </urn:PortletDescription>
  </urn:offeredPortlets>
</urn:getServiceDescriptionResponse
```

The `portletHandle` maps to the Static ID of the report region or the `reportid` parameter in the REST request. The `groupID` maps to the page id or the page parameter in the REST request. Finally, any parameters used by the SQL report are listed as children of the `keywords` node.

17.4 Accessing Data with Database Links

This section describes how to access data by creating a database link.

- [About Database Links](#) (page 17-31)
- [Creating a Database Link](#) (page 17-31)
- [Viewing an Existing Database Link](#) (page 17-31)

 **See Also:**

- "Managing Database Objects with Object Browser" in *Oracle Application Express SQL Workshop Guide*
- "Database Links" in *Oracle Database Administrator's Guide*

17.4.1 About Database Links

Because the Workspace home page runs in an 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*.

17.4.2 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.

17.4.3 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.

17.5 About DML Locking

This section describes how to use automatic Data Manipulation Language (DML) in Oracle Application Express to update or delete rows of a table.

- [About DML Locking](#) (page 17-32)
- [APEX_DML_LOCK_WAIT_TIME](#) (page 17-32)
- [FSP_DML_LOCK_ROW](#) (page 17-32)

17.5.1 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

17.5.2 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.

17.5.3 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.

18

Extending Application Capabilities

This section provides information about how to extend application capabilities including managing email, running background PL/SQL, utilizing plug-ins, and understanding jQuery support.

Topics:

- [Sending Email from an Application](#) (page 18-1)
- [Implementing Plug-ins](#) (page 18-3)
- [Understanding jQuery and jQuery UI Support](#) (page 18-13)
- [Manually Refreshing Oracle Application Express Components](#) (page 18-19)



See Also:

- [Oracle Application Express API Reference](#)
- ["Deploying an Application](#) (page 23-1)"

18.1 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](#) (page 18-1)
- [About Email Provisioning](#) (page 18-2)
- [About the Mail Queue](#) (page 18-3)
- [About the APEX_MAIL Package](#) (page 18-3)

18.1.1 About Configuring Email

To enable Oracle Application Express to send email, an Instance administrator must configure email settings. This section describes how to configure email settings in a full development environment and a runtime environment.

- [About Enabling Network Services](#) (page 18-2)
- [About Configuring Email in a Full Development Environment](#) (page 18-2)



See Also:

"Configuring Email" in *Oracle Application Express Administration Guide*

18.1.1.1 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_050100` 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*

18.1.1.2 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*

18.1.2 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*

18.1.3 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*

18.1.4 About the APEX_MAIL Package

You can send email from an Oracle Application Express application using the `APEX_MAIL` package. This 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*

18.2 Implementing Plug-ins

This section describes how to add, manage, reuse, and share plug-ins. You can find plug-in examples on the Oracle Application Express Plug-in Repository on the Oracle Technology Network and in the *Sample Database Application*.

- [About Plug-ins](#) (page 18-4)
- [Viewing Plug-in Examples](#) (page 18-4)
- [Accessing the Plug-ins Page](#) (page 18-5)

- [Creating a Plug-in](#) (page 18-5)
- [Editing a Plug-in](#) (page 18-7)
- [Adding Custom Attributes to a Plug-in](#) (page 18-7)
- [Uploading Files Associated with a Plug-in](#) (page 18-8)
- [Adding Events to a Plug-in](#) (page 18-10)
- [Deleting a Plug-in](#) (page 18-10)
- [Viewing the Plug-in Repository](#) (page 18-11)
- [Importing a Plug-in from the Plug-in Page](#) (page 18-11)
- [Exporting a Plug-in from the Plug-in Page](#) (page 18-12)
- [Resetting the Plug-in Interactive Report](#) (page 18-12)
- [Viewing Plug-in Utilization Page](#) (page 18-12)
- [Viewing Plug-in History](#) (page 18-13)

 **See Also:**

- ["Viewing Plug-in Examples](#) (page 18-4)"
- ["Viewing the Plug-in Repository](#) (page 18-11)"

18.2.1 About Plug-ins

Plug-ins enable developers to declaratively extend, share, and reuse the built-in types available with Oracle Application Express.

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.

18.2.2 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](#) (page 18-11)."
- Install the packaged application, *Sample Database Application*.

See "[Managing Packaged Applications in a Full Development Environment \(page 3-2\)](#)" and "[Installing and Running Sample Database Application \(page 3-8\)](#)."

- Go to the Oracle Learning Library at <http://www.oracle.com/oll/apex>. Enter search criteria in the field provided and click **Search**.

18.2.3 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.

See Also:

"[Customizing Interactive Reports in a Running Application \(page 8-92\)](#)"

18.2.4 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 \(page 1-27\)](#)."

1. Navigate to the Plug-ins page.
2. Click **Create**.

The Create Plug-in wizard appears.
3. For Create Plug-in, select the method by which you would like to create a plug-in and click **Next**.
4. 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.

11. For Help Text, enter help text used by the user to understand how the plug-in works.
12. For Comments, enter comments and notes that never display when the application is running.

To learn more about each option, see field-level Help.

13. 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:

- ["Accessing the Plug-ins Page \(page 18-5\)"](#)
- ["Adding Custom Attributes to a Plug-in \(page 18-7\)"](#)
- ["Uploading Files Associated with a Plug-in \(page 18-8\)"](#)
- ["Adding Events to a Plug-in \(page 18-10\)"](#)

18.2.5 Editing a Plug-in

To edit a plug-in:

1. Navigate to the Plug-ins page.
2. Click the plug-in you want to edit or view.
The Plug-in Create/Edit page appears.
3. Make modifications.
To learn more about each option, see field-level Help.
4. Click **Apply Changes**.

See Also:

- ["Accessing the Plug-ins Page \(page 18-5\)"](#)
- ["Creating a Plug-in \(page 18-5\)"](#)
- ["Adding Custom Attributes to a Plug-in \(page 18-7\)"](#)
- ["Uploading Files Associated with a Plug-in \(page 18-8\)"](#)
- ["Adding Events to a Plug-in \(page 18-10\)"](#)

18.2.6 Adding Custom Attributes to a Plug-in

A plug-in attribute is used to prompt the developer for additional data in the App Builder when the plug-in is used.

To add custom attributes to the plug-in:

1. Navigate to the Plug-ins page.
2. Click the plug-in you want to modify.
The Plug-in Create/Edit page appears.
3. 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.
4. 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.
5. 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.

 **See Also:**

["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.7 Uploading Files Associated with a Plug-in

This section describes how to upload the files associated with a plug-in.

- [Uploading a File to Associate with a Plug-in \(page 18-8\)](#)
- [Automatically Loading CSS and JavaScript Files \(page 18-9\)](#)

18.2.7.1 Uploading a File to Associate with a Plug-in

To upload a file:

1. Navigate to the Plug-ins page.
2. Select the plug-in.
The Plug-in Create/Edit page appears.
3. 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.
4. 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.
5. Click **Apply Changes**.

**See Also:**

"[Accessing the Plug-ins Page](#) (page 18-5)"

18.2.7.2 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 Plug-ins page.
2. Select the plug-in.
The Plug-in Create/Edit page appears.
3. 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.

4. Click **Apply Changes**.

**See Also:**

["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.8 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 Plug-ins page.
2. Click the plug-in you want to edit.
The Plug-in Create/Edit page appears
3. Under Events and click **Add Event**.
A new row displays under Events.
4. 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`.
5. Click **Add Event**.
6. Repeat steps 3 through 4 to add another event.
7. Click **Apply Changes**.

**See Also:**

["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.9 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 Plug-ins page.
2. Click the plug-in you want to delete.
The Plug-in Create/Edit page appears.
3. Click **Delete**.
4. To confirm, click **OK**.

**See Also:**

["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.10 Viewing the Plug-in Repository

The Plug-in Repository provides a central location where developers can share and download plug-ins. The repository is located on the Oracle Technology Network.

To view the Plug-in repository:

1. Navigate to the Plug-ins page. See ["Accessing the Plug-ins Page \(page 18-5\)"](#).
2. Click **View Plug-in Repository**.

The Oracle Application Express Plug-in Repository displays.

18.2.11 Importing a Plug-in from the Plug-in Page

Use this option to import an exported plug-in to your application. Importing a plug-in can be done from the Plug-ins page under Shared Components, as described here, or from the App Builder home page.

To import a plug-in:

1. Navigate to the Plug-ins page.
2. Click **Import**.
The import Plug-in page appears.
3. For Specify File:
 - a. Import file - Enter or browse to the name of the import file.
 - b. File Type - Select **Plug-in**.
 - c. File Character Set - Select the import file character set encoding.
 - d. Click **Next**.
4. For File Import Confirmation, click **Next**.
5. For Install, click **Install Plug-in**.

**See Also:**

- ["Importing Plug-ins \(page 23-33\)"](#)
- ["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.12 Exporting a Plug-in from the Plug-in Page

Use this option to export a plug-in definition to a file. This file can be imported into any APEX application. Exporting a plug-in can be done from the Plug-ins page under Shared Components, as described here, or from the workspace home page.

To export a plug-in from the Plug-in page:

1. Navigate to the Plug-ins page.
2. Under Tasks, click **Export Plug-in**.
The Export Plug-in page appears.
3. On the Export Plug-in page:
 - a. Application - Select the application to export the plug-in from.
 - b. Plug-in - Select plug-in.
 - c. File Format - Select file format of the plug-in export.
 - d. Click **Export**
4. Click **Export**.
A downloads complete message appears.

See Also:

- ["Exporting Plug-ins \(page 23-26\)"](#)
- ["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.13 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 Plug-ins page. See
2. On the Plug-ins page, click **Reset**.

See Also:

- ["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.14 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 Plug-ins page.
2. Click **Utilization**.

The Utilization page appears.

 **See Also:**

["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.2.15 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 Plug-ins page.
2. Click **History**.

The History page appears.

 **See Also:**

["Accessing the Plug-ins Page \(page 18-5\)"](#)

18.3 Understanding jQuery and jQuery UI Support

Oracle Application Express includes the jQuery 1.11.3 and 2.2.3, jQuery UI 1.10.4, and jQuery Mobile - 1.4.5 JavaScript libraries. This section discusses the features available in jQuery UI and best practices when referencing jQuery libraries in your JavaScript code.

- [About Compatibility Issues When Upgrading from Application Express Release 4.X to 5.1 \(page 18-13\)](#)
- [About Available jQuery UI Features \(page 18-15\)](#)
- [Referencing the jQuery Library in Your JavaScript Code \(page 18-16\)](#)

18.3.1 About Compatibility Issues When Upgrading from Application Express Release 4.X to 5.1

From Oracle Application Express Release 4.X to 5.1, the JavaScript libraries for jQuery and jQueryUI used by Oracle Application Express have been updated as follows:

- jQuery has been updated from version 1.7.1 to version 2.2.3.

- jQueryUI has been updated from version 1.8.14 to version 1.10.4.

 **Tip:**

For a list of bug fixes and enhancement, see the jQuery and jQueryUI release notes at:

- http://docs.jquery.com/Downloading_jQuery#Current_Release
- <http://jqueryui.com/docs/Changelog>

This upgrade may impact existing applications in the following ways.

Possible Impacts of Updated jQuery Version

If you have used jQuery in your own JavaScript code in applications, then you may be impacted by changed behavior caused by the updated jQuery version. The most significant impact concerns the use of the `jQuery.attr` method. This method returns slightly different values in some circumstances. Please review the jQuery change log for further details of this and other changes.

If you are concerned this may impact your applications, Oracle suggests the following:

1. Fallback to the previous version of jQuery for your code. You can include jQuery 1.7.1 and jQuery UI 1.8.14 by adding the following to your Page Template > Header, within the `<head>...</head>` tags, importantly ensuring this is done directly after the `#HEAD#` substitution string and before you include any of your libraries:

```
...
#HEAD#
<script src="#IMAGE_PREFIX#libraries/jquery/1.7.1/jquery-1.7.1.min.js"
type="text/javascript"></script>
...
```

Oracle Application Express still includes jQuery 2.1.3 as part of the `#APEX_JAVASCRIPT#` substitution, but including 1.7.1 afterwards means that the variables `$` and `jQuery` will now point to jQuery version 1.7.1, which you can continue to use in your applications.

 **Tip:**

jQuery 1.7.1 is still included with Oracle Application Express distribution, so you do not need to add this to your web server.

2. Review and retest your jQuery code to see if you are impacted by the changes. This is more work initially, but has two advantages. First, you only need to load one version of the jQuery library into your application. Second, you also benefit from the enhancements included with the most recent version.

Possible Impacts of Updated jQuery UI

You may experience compatibility issues if you have included components of jQuery UI that are not included by default with Oracle Application Express. To reduce page size and processing, only the components of jQuery UI used by Oracle Application Express base functionality are included with Oracle Application Express. All of these

default components now use jQuery UI 1.10.4.. You could experience compatibility issues if you are referencing an older version of jQuery UI.

All of these default components are now using jQuery UI 1.10.4., whereas any components you may have previously included may be referencing an older version. This means there may be compatibility issues between the versions of jQuery UI components you have included and the updated jQuery and jQuery UI versions we include by default.

If you are concerned this may impact your applications, Oracle suggests two possible options:

1. Fallback to the previous versions of jQuery and jQuery UI in your code. You can include jQuery 1.7.1 and jQuery UI 1.8.14 by adding the following to your Page Template > Header, within the <head>...</head> tags, importantly ensuring this is done straight after the #HEAD# substitution string and before you include any of your libraries:

```
...
#HEAD#
<script src="#IMAGE_PREFIX#libraries/jquery/1.7.1/jquery-1.7.1.min.js"
type="text/javascript"></script>
<script src="#IMAGE_PREFIX#libraries/jquery-ui/1.8.14/ui/minified/jquery-
ui-1.8.14.custom.min.js" type="text/javascript"></script>
...
```

Although Oracle Application Express still includes jQuery 2.1.3 and jQuery UI 1.10.4 as part of the #APEX_JAVASCRIPT# substitution, Oracle just includes the older versions for compatibility with your other jQuery UI components you may be referencing.

Tip:

jQuery 2.1.3 and jQuery UI 1.10.4. are still included with Oracle Application Express, so you do not need to add this to your web server.

2. Review your jQuery UI code, update any jQuery UI components to reference version 1.10.4., and retest. This approach is more work initially, but has the advantage that you only need to load one version of the jQuery library and one version of the jQuery UI library into your application. You also benefit from the enhancements included with the most recent versions.

To make the reference agnostic to future version changes, Oracle suggests you search the files containing your JavaScript code and also in Application Search in the App Builder for the following `libraries/jquery-ui/` and replace it with `#JQUERYUI_DIRECTORY#`.

18.3.2 About Available jQuery UI Features

Oracle Application Express only loads the components of jQuery UI that are required for base Oracle Application Express functionality. Oracle Application Express does not include the entire jQuery UI library since doing so would significantly add to download and processing time for each page load. Oracle Application Express includes these components by default:

- jQuery UI Core - Required for all interactions and widgets.

- Query UI Widget - The widget factory that is the base for all widgets.
- jQuery UI Mouse - The mouse widget is a base class for all interactions and widgets with heavy mouse interaction.
- jQuery UI Position - A utility plug-in for positioning elements relative to other elements.
- jQuery UI Draggable - Makes any element on the page draggable.
- jQuery UI Resizable - Makes any element on the page resizable.
- jQuery UI Dialog - Opens existing mark up in a draggable and resizable dialog.
- jQuery UI Datepicker - A datepicker than can be toggled from a input or displayed inline.
- jQuery UI Effects - Extends the internal jQuery effects, includes morphing, easing, and is required by all other effects.
- jQuery UI Effects Drop - A drop out effect by moving the element in one direction and hiding it at the same time.
- jQuery UI Tooltip - Shows additional information for any element on hover or focus

For more information about jQuery UI and these specific components, see the jQuery UI site:

<http://jqueryui.com/>

- [About Accessing Other jQuery UI Components](#) (page 18-16)

18.3.2.1 About Accessing Other jQuery UI Components

Oracle Application Express does not include the entire jQuery UI library. You can easily activate other components of jQuery UI by just including the relevant files. For example, to include the Tabs jQuery UI Widget in your application, you would include the following in your User Interface definition.

Set Cascading Style Sheets to:

```
#JQUERYUI_DIRECTORY#themes/base/#MIN_DIRECTORY#jquery.ui.tabs#MIN#.css
```

Set JavaScript to:

```
#JQUERYUI_DIRECTORY#ui/#MIN_DIRECTORY#jquery.ui.tabs#MIN#.js
```

Tip:

You do not need to include the Dependencies for Tabs of UI Core and UI Widget since these are included by default with Oracle Application Express, as shown in the previous list of default components.

18.3.3 Referencing the jQuery Library in Your JavaScript Code

Determining when to use the `$`, `jQuery`, and `apex.jQuery` references to the jQuery library in your own JavaScript code depends on where you use it.

Newer versions of Oracle Application Express may include updated versions of the jQuery libraries. When new versions of jQuery are released, they sometimes introduce changes that could break existing functionality. These issues are documented in the Change Log for a particular version's Release Notes. To minimize your risk when upgrading to a newer version of Oracle Application Express, Oracle recommends the best practices described in the following sections.

- [About Managing JavaScript Code in Your Application](#) (page 18-17)
- [About Using JavaScript Code in a Plug-In](#) (page 18-18)

18.3.3.1 About Managing JavaScript Code in Your Application

If you want to use jQuery in your own JavaScript code in an application, Oracle recommends you use `jQuery` or the shortcut `$` for the following reasons:

1. If you upgrade Oracle Application Express, there is the potential that the newer jQuery version may break your existing code.

To avoid intensive testing and rewriting of your application to the new jQuery version, you can include the old jQuery version that was shipped with Oracle Application Express release 4.1 in your Page Template, Header within the `<head>...</head>` tags after the `#HEAD#` substitution string. Consider the following example:

```
...
#HEAD#
...
<script src="#IMAGE_PREFIX#libraries/jquery/1.6.2/jquery-1.6.2.min.js"
type="text/javascript"></script>
...
```

In this example, the references to the jQuery library '\$' and 'jQuery' point to jQuery version 1.6.2 (the version included in Oracle Application Express release 4.1) and not the newer version 2.1.3 included with Oracle Application Express release 5.0.

 **Tip:**

Oracle Application Express release 5.0 includes jQuery version 2.1.3 so you do not need to add this to your web server.

Note that no additional code changes necessary. This approach minimizes the risk of breaking custom jQuery code when upgrading to a newer version of Application Express.

2. If jQuery plug-in you use requires a different version of jQuery and you want to use that version in other code in your application.

In this case, you can include version of jQuery library (the newer or older) into your page template as described in the previous step (that is, after the `#HEAD#` substitution string). The variables `$` and `jQuery` point to the appropriate jQuery version.

3. If a jQuery plug-in you use requires a different version of jQuery and you do not want to use that version in other code in an application.

For example, suppose a jQuery plug-in requires jQuery 1.6.2, but you want to use jQuery 1.7.1 which included with Oracle Application Express. Include jQuery version 1.6.2 into your Page Template, Header, within the `<head>...</head>` tags, before the `#HEAD#` substitution string.

In this case, you also need a JavaScript snippet to define a new variable to store the plug-in specific jQuery version (for example, `jQuery_1_6_2`) and then assign it. Consider the following example:

```
...
<script src="#IMAGE_PREFIX#libraries/jquery/1.6.2/jquery-1.6.2.min.js"
type="text/javascript"></script>
<script type="text/javascript">
    var jQuery_1_6_2 = jQuery;
</script>
...
#HEAD#
...
```

You must also modify the initialization code of the jQuery plug-in to use the `jQuery_1_6_2` variable, for example:

```
(function($) {
    ... plugin code ...
})(jQuery_1_6_2);
```

For more information about jQuery plug-in initialization, see ["About Using JavaScript Code in a Plug-In \(page 18-18\)."](#)

 **Tip:**

In all of these scenarios, use the variables `$` and `jQuery` in your application to specify the version you require. Note that the variable `apex.jQuery` still points to the version that ships with Oracle Application Express to support built-in functionality, which depends on that jQuery version.

18.3.3.2 About Using JavaScript Code in a Plug-In

If you want to use jQuery in an Oracle Application Express plug-in, Oracle recommends using the `apex.jQuery` reference to the jQuery library. This reference should even include a modification of the initialization code of the included jQuery plug-in to use the `apex.jQuery` reference.

If you look at the JavaScript code of a jQuery plug-in, notice that most have the following code structure:

```
(function($) {
    ... plugin code ...
})(jQuery);
```

This structure declares an anonymous JavaScript function with a parameter `$`, which is immediately called and passed the parameter value of the current jQuery variable. The use of `$` as the jQuery shortcut is implemented in a safe manner without having to rely on the fact that `$` is still used to reference the main jQuery library.

Oracle recommends copying the jQuery plug-in file and prefixing it with `apex` (for example, `apex.jquery.maskedinput-1.2.2.js`). This makes it obvious to Oracle Application Express that the file has been modified. Then, change the reference from `jQuery` to `apex.jQuery` in the initialization code as shown in the following example.

```
(function($) {  
    ... plugin code ...  
})(apex.jQuery);
```

Other Reasons to Use this Approach

As a plug-in developer, you want to minimize your testing effort and create an environment where you have full control. Assume you are not following the recommendations described in the previous section. What happens if your plug-in is using a different version of jQuery?

This may result in strange plug-in behavior. Furthermore, it may be difficult to reproduce that behavior because you will not know if your plug-in is actually referencing a different version of jQuery. Using the `apex.jQuery` namespace reduces your risk and also the risk to your users. This approach enables you to test your plug-ins with the Oracle Application Express versions you want to support.

18.4 Manually Refreshing Oracle Application Express Components

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" );
```

19

Managing Application Security

While administrators are primarily responsible for ensuring the security of the Oracle Application Express installation, developers are responsible for building secure applications. This section describes how to implement best practices for Oracle Application Express application security.

Topics:

- [Understanding Administrator Security Best Practices](#) (page 19-1)
- [Understanding Developer Security Best Practices](#) (page 19-18)
- [Controlling Access to Applications, Pages, and Page Components](#) (page 19-39)
- [Establishing User Identity Through Authentication](#) (page 19-45)
- [Providing Security Through Authorization](#) (page 19-59)

See Also:

["Running Advisor to Check Application Integrity](#) (page 22-1)"

19.1 Understanding Administrator Security Best Practices

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).

This section describes security best practices for Oracle Application Express administrators.

- [About Oracle Application Express Administrator Roles](#) (page 19-2)
- [About Configuring Oracle REST Data Services with Oracle Application Express](#) (page 19-2)
- [About Configuring Oracle HTTP Server with mod_plsql with Oracle Application Express](#) (page 19-3)
- [About Security Considerations When Using the Embedded PL/SQL Gateway](#) (page 19-4)
- [Managing Instance Security](#) (page 19-4)

- [Configuring Instance Settings](#) (page 19-10)
- [About Configuring Workspace Purge Settings](#) (page 19-12)
- [Understanding Workspace Management](#) (page 19-12)
- [About Integrating with Oracle BI Publisher](#) (page 19-14)
- [About the Advantages of the Application Express Runtime Environment](#) (page 19-14)
- [Enabling Network Services in Oracle Database 11g or Later](#) (page 19-14)
Learn about enabling network services in Oracle Database 11g or later versions.



See Also:

["Understanding Developer Security Best Practices](#) (page 19-18)"

19.1.1 About Oracle Application Express Administrator Roles

To access the Oracle Application Express development environment, users log in to a shared work area called a workspace. Oracle Application Express includes two different types of administrators:

- **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*

19.1.2 About Configuring 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*

19.1.3 About Configuring Oracle HTTP Server with `mod_plsql` with Oracle Application Express

 **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.

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.

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 `wwv_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*

19.1.4 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.

19.1.5 Managing Instance Security

This section provides a summary of security best practices when managing an Oracle Application Express instance.

- [About Creating Login Controls](#) (page 19-5)
- [About Enabling Public File Upload](#) (page 19-5)
- [About Restricting User Access by IP Address](#) (page 19-5)
- [About Specifying an Instance Proxy](#) (page 19-6)
- [About Utilizing Secure Sockets Layer](#) (page 19-6)
- [About Enabling RESTful Access](#) (page 19-7)
- [About Rejoin Sessions](#) (page 19-7)
- [About Isolating Workspaces](#) (page 19-8)
- [About Utilizing Session Timeout](#) (page 19-8)
- [Restricting Password Reuse](#) (page 19-9)
- [About Using SQL in Websheets](#) (page 19-10)
- [About Enabling RESTful Services](#) (page 19-10)

 **See Also:**

"Configuring Security Settings" in *Oracle Application Express Administration Guide*

19.1.5.1 About Creating Login Controls

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 for an entire instance or for each individual workspace. 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:**

"Creating Login Controls for an Instance" and "Creating Account Login Controls for a Workspace" in *Oracle Application Express Administration Guide*.

19.1.5.2 About Enabling Public File Upload

The Allow Public File Upload attribute enables administrators 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*

19.1.5.3 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*

19.1.5.4 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*

19.1.5.5 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:**

"Requiring HTTPS" in *Oracle Application Express Administration Guide*

19.1.5.6 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:

"Controlling RESTful Services for an Instance" in *Oracle Application Express Administration Guide*

19.1.5.7 About Rejoin Sessions



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](#) (page 19-8)."

The Rejoin Sessions attribute controls if Oracle Application Express supports application URLs that do not contain session IDs. 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 \(page 2-12\)"](#)
- ["Browser Security \(page 6-30\)"](#)
- ["Configuring Rejoin Sessions in Page Designer \(page 7-34\)"](#)
- ["Configuring Rejoin Sessions in Component View \(page 24-26\)"](#)
- "Configuring Rejoin Sessions for an Instance" in *Oracle Application Express Administration Guide*

19.1.5.8 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*

19.1.5.9 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 (page 6-26)" 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*.

19.1.5.10 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_050100;
```

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*

19.1.5.11 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*

19.1.5.12 About Enabling RESTful Services

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.

 **See Also:**

"Controlling RESTful Services for an Instance" in *Oracle Application Express Administration Guide*

19.1.6 Configuring Instance Settings

This section provides a summary of security best practices when configuring an Oracle Application Express instance.

- [About Enabling Self Service Workspace Provisioning](#) (page 19-11)
- [About Enabling and Configuring Email](#) (page 19-11)
- [About Configuring Storage Options](#) (page 19-12)
- [About Creating Wallets](#) (page 19-12)

 **See Also:**

"Configuring Instance Settings" in *Oracle Application Express Administration Guide*

19.1.6.1 About Enabling Self Service Workspace Provisioning

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*

19.1.6.2 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](#) (page 19-15)"

19.1.6.3 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*

19.1.6.4 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*

19.1.7 About Configuring Workspace Purge Settings

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*

19.1.8 Understanding Workspace Management

This section provides a summary of security best practices when managing workspaces in Oracle Application Express.

- [About Managing Workspace to Schema Assignments](#) (page 19-13)
- [About Managing Developers and Users](#) (page 19-13)
- [About Managing Component Availability](#) (page 19-13)

 **See Also:**

"Managing Existing Workspaces" in *Oracle Application Express Administration Guide*

19.1.8.1 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*

19.1.8.2 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" and "Managing Users in a Workspace" in *Oracle Application Express Administration Guide*

19.1.8.3 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*

19.1.9 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" and "Configuring Report Printing" in *Oracle Application Express Administration Guide*

19.1.10 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*

19.1.11 Enabling Network Services in Oracle Database 11g or Later

Learn about enabling network services in Oracle Database 11g or later versions.

- [When and Why Network Services Must be Enabled](#) (page 19-15)
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](#) (page 19-15)
Demonstrates how to grant connect privileges to any host for the `APEX_050100` database user.
- [Granting Connect Privileges in Oracle Database 12c](#) (page 19-17)
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](#) (page 19-17)
Learn how to identify a invalid ACL error by running the query.

19.1.11.1 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 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_050100` 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*.

19.1.11.2 Granting Connect Privileges Prior to Oracle Database 12c

Demonstrates how to grant connect privileges to any host for the `APEX_050100` database user.

The following example demonstrates how to grant connect privileges to any host for the `APEX_050100` 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_050100
  -- the "connect" privilege if APEX_050100 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_050100',
    'connect') IS NULL THEN
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(ACL_PATH,
    'APEX_050100', 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_050100', 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_050100
  -- the "connect" privilege if APEX_050100 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_050100',
    'connect') IS NULL THEN
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(ACL_PATH,
    'APEX_050100', 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_050100', TRUE, 'connect');
    DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL('local-access-users.xml', 'localhost');
END;
/
COMMIT;
```

19.1.11.3 Granting Connect Privileges in Oracle Database 12c

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_050100` 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_050100',
                      principal_type => xs_acl.p_type_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_050100',
                      principal_type => xs_acl.p_type_db));
END;
/
```

19.1.11.4 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_050100` user.

19.2 Understanding Developer Security Best Practices

This section describes security best practices for Oracle Application Express developers.

- [About Items of Type Password](#) (page 19-19)
- [Identifying At Risk Password Items](#) (page 19-19)
- [Understanding Cross-Site Scripting Protection](#) (page 19-21)
- [About Session State and Security](#) (page 19-26)
- [Understanding Session State Protection](#) (page 19-26)
- [About Securing File Uploads](#) (page 19-39)

See Also:

- ["Understanding Administrator Security Best Practices](#) (page 19-1)"
- ["About Oracle Application Express Administrator Roles](#) (page 19-2)"

19.2.1 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. This prevents 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](#) (page 19-26)".

See Also:

- "[About Item Types](#) (page A-1)"
- "[Editing Page-Level Items in Page Designer](#) (page 13-14)"
- "[Creating Page-Level Items in Component View](#) (page 13-22)"

19.2.2 Identifying 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 the value in session state un-encrypted.

This section explains how to use the Security Profiles and Password Items reports to help identify at risk password items.

- [Viewing the Security Profiles Report](#) (page 19-20)
- [Viewing the Password Items Report](#) (page 19-20)

19.2.2.1 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

19.2.2.2 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.

19.2.3 Understanding Cross-Site Scripting Protection

This section explains best practices for protecting your application from a cross site-scripting security breach.

- [What Is Cross Site-scripting Security Breach?](#) (page 19-21)
- [Protecting HTML Regions and Other Static Areas](#) (page 19-21)
- [About Protecting Dynamic Output](#) (page 19-23)
- [About Protecting Report Regions](#) (page 19-24)
- [About Protecting Form Items](#) (page 19-25)
- [About Restricting Characters Entered on an Item](#) (page 19-25)

19.2.3.1 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.

19.2.3.2 Protecting HTML Regions and Other Static Areas

This section explains how to protect HTML regions and other static display areas.

- [About Protecting HTML Regions and Other Static Areas](#) (page 19-21)
- [About Safe Item Display Types](#) (page 19-22)
- [About the Rules Used to Determine Whether to Escape Values](#) (page 19-22)
- [About Using Safe Item Types to Hold and Emit HTML Markup](#) (page 19-23)

19.2.3.2.1 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 \(page 2-26\)"](#)
- ["Controlling Output Escaping in Substitution Strings \(page 2-27\)"](#)

19.2.3.2.2 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 \(page 13-1\)"](#)

19.2.3.2.3 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.

19.2.3.2.4 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.

19.2.3.3 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 \(page 19-22\)"](#)

19.2.3.4 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 \(page 2-27\)"](#)
- ["Editing Interactive Report Column Attributes in Page Designer \(page 8-40\)"](#)
- ["Editing Column Attributes in Component View \(page 8-67\)"](#)

19.2.3.5 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 \(page 2-27\)"](#)

19.2.3.6 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 \(page 13-16\)"](#)

19.2.4 About 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.

A few best practices for developers 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.

19.2.5 Understanding 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.

- [How Session State Protection Works](#) (page 19-27)

- [Enabling Session State Protection](#) (page 19-27)
- [Configuring Session State Protection](#) (page 19-29)

19.2.5.1 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.

19.2.5.2 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](#) (page 19-27)
- [Enabling Session State Protection from Session State Protection](#) (page 19-28)

19.2.5.2.1 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](#) (page 19-28)

19.2.5.2.1.1 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.

19.2.5.2.2 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.

19.2.5.3 Configuring Session State Protection

 **Tip:**

Before you can configure security attributes, you must first enable Session State Protection. See "[Enabling Session State Protection](#) (page 19-27)".

- [About Configuring Session State Protection](#) (page 19-29)
- [Reviewing Existing Session State Protection Settings](#) (page 19-29)
- [Configuring Session State Protection Using a Wizard](#) (page 19-30)
- [Configuring Session State Protection for Pages](#) (page 19-33)
- [Configuring Session State Protection for Page Items](#) (page 19-35)
- [Configuring Session State Protection for Application Items](#) (page 19-37)

19.2.5.3.1 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.

19.2.5.3.2 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

19.2.5.3.3 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**.

19.2.5.3.4 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 \(page 8-93\)"](#)

19.2.5.3.5 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 \(page 8-93\)"](#)

19.2.5.3.6 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 \(page 8-93\)"](#)

19.2.6 About 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. For this reason, Oracle recommends that developers either use the methods described in "[Understanding BLOB Support in Forms and Reports](#) (page 8-130)" or use the methods described in Oracle Application Express How To Documents for file upload on OTN at:

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

Either of these methods results in the uploaded files being stored in schema tables and not the public table.

See Also:

- "[About the Differences Between Page Items and Application Items](#) (page 13-2)"
- "[About Item Types](#) (page A-1)" to learn more about creating a File Browse page-level item

19.3 Controlling Access to Applications, Pages, and Page Components

You can control access to an application, individual pages, or page components by creating an access control list. To create an access control page, the application schema must have `CREATE TABLE`, `CREATE TRIGGER`, `CREATE SEQUENCE` privileges.

- [How the Access Control List Works](#) (page 19-40)
- [Creating an Access Control List](#) (page 19-40)
- [Selecting an Application Mode and Adding Users](#) (page 19-42)
- [About Controlling Access for Pages and Page Components](#) (page 19-44)

See Also:

"Building Queries with Query Builder" in *Oracle Application Express SQL Workshop Guide*

19.3.1 How the Access Control List Works

You create an access control list by running the Access Control Wizard. The Access Control Wizard creates a page named **Access Control Administration** which contains a list of application modes and an Access Control List. Once you create the Access Control Administration page, you must:

- **Step 1** - Run the Access Control Administration page.
- **Step 2** - Select one of the following application modes:
 - Full access to all, access control list not used.
 - Restricted access. Only users defined in the access control list are allowed.
 - Public read only. Edit and administrative privileges controlled by access control list.
 - Administrative access only.
- **Step 3** - Add users to the Access Control List.

In addition to creating the Access Control Administration page, the Access Control Wizard also creates:

- Two tables within the application's default schema to manage the access control
- The authorization schemes that correspond to the application mode list options
- The privileges available in the Access Control List

You can control access to a specific page or page component by selecting one of these authorization schemes on the page or component attributes pages. Once you create an Access Control, you can customize the page, tables and values to suit the specific needs of your application.



See Also:

["Attaching an Authorization Scheme to an Application \(page 19-64\)"](#)

19.3.2 Creating an Access Control List

You create an access control list by creating a page. You can create a page on the Application home page or while viewing a page.

- [Creating an Access Control from the Application Home Page \(page 19-40\)](#)
- [Creating an Access Control While Editing a Page \(page 19-41\)](#)

19.3.2.1 Creating an Access Control from the Application Home Page

To create an access control list from the Application home page:

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

The Application home page appears.

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 **Access Control**.
 - c. Click **Next**.
5. For Pages:
 - a. Administration Page Number - Specify an unused page number.
 - b. Page Mode - Select a page mode. To learn more, see field-level Help.
 - c. Page Group - Select a page group.
 - d. Click **Next**.
6. 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.
7. Review the confirmation page and click **Create**.

19.3.2.2 Creating an Access Control While Editing a Page

To create an access control list while editing a page:

1. Navigate to the appropriate page.
2. Access the Create Page Wizard:
 - Page Designer - Click the **Create** menu at the top of the page and select **Page**.
 - Component View - Click the **Create** menu at the top of the page and select **New Page**.
3. For Create a Page:
 - a. User Interface - Select a user interface for the page.
 - b. Select a page type - Select **Access Control**.
 - c. Click **Next**.
4. Click **New page** and select **New Page**.
5. For page type, select **Access Control** and click **Next**.

The Access Control Wizard appears.
6. For Pages:
 - a. Administration Page Number - Specify an unused page number.
 - b. Page Mode - Select a page mode. To learn more, see field-level Help.
 - c. 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.
8. Review the confirmation page and click **Create**.

19.3.3 Selecting an Application Mode and Adding Users

You can control access to an application by running the Access Control Administration page, selecting an application mode, and then adding users to the Access Control list.

- [Selecting an Application Mode](#) (page 19-42)
- [Adding Users to an Access Control List and Selecting Privileges](#) (page 19-43)
- [Removing Users from the Access Control List](#) (page 19-44)

19.3.3.1 Selecting an Application Mode

To select an application mode:

1. Create an access control list.
The wizard creates a page named Access Control Administration.
2. Run the Access Control Administration page.
3. For Application Mode, select an option:
 - **Full access to all, access control list not used.**
Select this option to enable all users access to an application.
 - **Restricted access. Only users defined in the access control list are allowed.**
Select this option to restrict access to users on the Access Control List. Only users on the Access Control List can view pages and components associated with an authorization scheme.
 - **Public read only. Edit and administrative privileges controlled by access control list.**
Provides public access to pages and components associated with the access control - view authorization scheme.
 - **Administrative access only.**
Only users with Administrator privileges can access pages or components associated with an authorization scheme.
4. Click **Set Application Mode**.

Application Administration

Application Mode **Full access to all, access control list not used.** ?

Restricted access. Only users defined in the access control list are allowed.

Public read only. Edit and administrative privileges controlled by access control list.

Administrative access only.

Access Control List

Identify usernames which correspond to this application's authentication scheme.

Find ?

<input type="checkbox"/>	Username ↑	Privilege	Last Changed By	Date
No data found.				

The user interface of your page depends on the theme you selected for your application.

Next, add users to the Access Control List.

See Also:

- ["Creating an Access Control List \(page 19-40\)"](#)
- ["Running an Application or Page \(page 2-7\)"](#)
- ["Using Themes \(page 11-1\)"](#)

19.3.3.2 Adding Users to an Access Control List and Selecting Privileges

To add users to the Access Control List:

1. Create an access control list.
The wizard creates a page named Access Control Administration.
2. Run the Access Control Administration page.
3. Under Access Control List, click **Add User**.
A new row appears.

Access Control List

Identify usernames which correspond to this application's authentication scheme.

Find

<input type="checkbox"/>	Username	Privilege	Last Changed By	Date
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	(null)	(null)

4. Enter a user in the Username field.
5. Associate a privilege with the user. Available options include:
 - **Administrator**
 - **Edit**
 - **View**
6. Click **Apply Changes**.
7. Repeat steps 1 to 5 for all users.



See Also:

- ["Creating an Access Control List \(page 19-40\)"](#)
- ["Running an Application or Page \(page 2-7\)"](#)

19.3.3.3 Removing Users from the Access Control List

To remove users from the Access Control List:

1. Select the user to be removed by selecting the check box to the left of the user name.
2. Click **Delete**.

19.3.4 About Controlling Access for Pages and Page Components

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 selecting one of the following authorization schemes on the page or component attributes pages:

- **Access control administrator.** Only users with Administrator privileges can view the page or component.

- **Access control - edit.** Users with both Edit and Administrator privileges can view the page or component. Users with View privileges cannot view the page or component.
- **Access control - view.** Users with Administrator, Edit, or View privileges can view the page or component.
- **Not access control administrator.** Users with Administrator privileges cannot view the page or component.
- **Not access control - edit.** Users with both Edit and Administrator privileges cannot view the page or component. Users with View privileges can view the page or component.
- **Not access control - view.** 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 \(page 19-63\)"](#)

19.4 Establishing User Identity Through Authentication

Authentication is the process of establishing each user's identity before they can access your application. This section describes how to use authentication to establish a user's identity.

- [Understanding Authentication \(page 19-45\)](#)
- [Creating an Authentication Scheme \(page 19-47\)](#)
- [Understanding Preconfigured Authentication Schemes \(page 19-48\)](#)
- [Managing Existing Authentication Schemes \(page 19-57\)](#)
- [Viewing the Authentication Scheme Associated with an Application \(page 19-58\)](#)
- [Changing the Authentication Scheme Associated with an Application \(page 19-58\)](#)
- [Creating a Login Page \(page 19-59\)](#)
- [Viewing Authentication Scheme Reports \(page 19-59\)](#)

19.4.1 Understanding Authentication

Authentication may require a user identify a user name and password or could involve the use of digital certificates or a secure key.

- [How Authentication Works \(page 19-46\)](#)
- [About Support for Deep Linking \(page 19-46\)](#)
- [About Determining Whether to Include Authentication \(page 19-47\)](#)

19.4.1.1 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" \(page 2-37\)](#)
- ["Configuring Security Attributes" \(page 6-23\)](#)

19.4.1.2 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.

19.4.1.3 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 \(page 19-48\)"](#)
- ["Changing the Authentication Scheme Associated with an Application \(page 19-58\)"](#)
- ["Creating and Editing an Authorization Scheme \(page 19-60\)"](#)

19.4.2 Creating an Authentication Scheme

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 \(page 19-48\)"](#)
- ["Custom Authentication \(page 19-49\)"](#)
- ["About the Search Bar \(page 8-93\)"](#)

19.4.3 Understanding 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 \(page 18-3\)."](#)

- [Application Express Accounts \(page 19-48\)](#)
- [Custom Authentication \(page 19-49\)](#)
- [Database Accounts \(page 19-51\)](#)
- [HTTP Header Variable \(page 19-52\)](#)
- [LDAP Directory Verification \(page 19-53\)](#)
- [No Authentication \(using DAD\) \(page 19-54\)](#)
- [Open Door Credentials \(page 19-55\)](#)
- [Oracle Application Server Single Sign-On Server \(page 19-56\)](#)

19.4.3.1 Application Express Accounts

This section explains when you might want to use the Application Express Accounts authentication scheme and explains how to configure it.

- [About Application Express Accounts \(page 19-48\)](#)
- [Setting Up Application Express Accounts Authentication \(page 19-49\)](#)

19.4.3.1.1 About 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.

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.

This 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*

19.4.3.1.2 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**.

19.4.3.2 Custom Authentication

This section explains when you might want to use the Custom Authentication and explains how to configure it.

- [About Custom Authentication](#) (page 19-49)
- [Setting Up Custom Authentication](#) (page 19-50)
- [About Session Management Security](#) (page 19-51)

19.4.3.2.1 About Custom Authentication

Creating an authentication scheme from scratch gives you complete control over your authentication interface. This 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](#) (page 18-3)."

19.4.3.2.2 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](#) (page 1-27)"

19.4.3.2.3 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.

19.4.3.3 Database Accounts

This section explains when you might want to use Database Account Credentials authentication and explains how to configure it.

- [About Database Account Credentials](#) (page 19-51)
- [Setting Up Database Account Credentials](#) (page 19-51)

19.4.3.3.1 About Database Account Credentials

Database Account Credentials utilizes database schema accounts. This authentication scheme requires that a database user (schema) exist in the local database. When using this method, the user name and password of the database account is used 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.

19.4.3.3.2 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**.

19.4.3.4 HTTP Header Variable

This section explains when you might want to use the HTTP Header Variable authentication and explains how to configure it.

- [About HTTP Header Variable](#) (page 19-52)
- [Setting Up HTTP Header Variable](#) (page 19-52)

19.4.3.4.1 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 this 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).

19.4.3.4.2 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](#) (page 1-27)"

19.4.3.5 LDAP Directory Verification

This section explains when you might want to use LDAP Credentials authentication and explains how to configure it.

- [About LDAP Directory Verification](#) (page 19-53)
- [Setting Up LDAP Directory Verification](#) (page 19-53)

19.4.3.5.1 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.

19.4.3.5.2 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 \(page 1-27\)"](#)
- ["Configuring Wallet Information" in *Oracle Application Express Administration Guide*](#)
- ["Using Oracle Wallet Manager" in *Oracle Database Advanced Security Guide*](#)

19.4.3.6 No Authentication (using DAD)

This section explains when you might want to use Database Access Descriptor (DAD) database authentication and explains how to configure it.

- [About DAD Credentials Verification \(page 19-55\)](#)
- [Setting Up DAD Credentials Verification \(page 19-55\)](#)

19.4.3.6.1 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.

19.4.3.6.2 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**.

19.4.3.7 Open Door Credentials

This section explains when you might want to use Open Door Credentials and explains how to configure it.

- [About Door Credentials](#) (page 19-56)

- [Setting Up Open Door Credentials](#) (page 19-56)

19.4.3.7.1 About Door Credentials

Open Door Credentials enables anyone to access your application using a built-in login page that captures a user name. This authentication method is useful during application development.

19.4.3.7.2 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**.

19.4.3.8 Oracle Application Server Single Sign-On Server

This section explains when you might want to use Oracle Application Server Single Sign-On verification and explains how to configure it.

- [About Application Server Single Sign-On Server](#) (page 19-56)
- [Setting Up Oracle Application Server Single Sign-On](#) (page 19-57)

19.4.3.8.1 About Application Server Single Sign-On Server

Oracle Application Server Single Sign-On verification 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.

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.

19.4.3.8.2 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 \(page 1-27\)"](#)

19.4.4 Managing Existing Authentication Schemes

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 \(page 8-93\)"](#)
- ["Viewing Field-Level Help \(page 1-27\)"](#)

19.4.5 Viewing the 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**.

19.4.6 Changing 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 \(page 19-48\)"](#)

19.4.7 Creating a 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**.

19.4.8 Viewing 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.

19.5 Providing Security Through Authorization

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](#) (page 19-60)

- [Creating and Editing an Authorization Scheme](#) (page 19-60)
- [Attaching an Authorization Scheme to an Application, Page, or Components](#) (page 19-63)
- [Viewing Authorization Reports](#) (page 19-65)

19.5.1 How Authorization Schemes Work

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.

19.5.2 Creating and Editing an Authorization Scheme

This section explains 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](#) (page 19-60)
- [Creating an Authorization Scheme](#) (page 19-61)
- [Editing Attributes of an Existing Authorization Scheme](#) (page 19-62)
- [Changing the Evaluation Point Attribute](#) (page 19-62)
- [About Resetting Authorization Scheme State](#) (page 19-63)

19.5.2.1 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 19-1 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

Table 19-1 (Cont.) Authorization Scheme Types

Authorization Scheme Types	Description
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.
Value of Item in Expression 1 Equals Expression 2	Enter an item name and a 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.

19.5.2.2 Creating an Authorization Scheme

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 a unique name that identifies this authorization scheme.
 - b. Scheme Type - Select how this authorization scheme will be applied. See "[About Authorization Scheme Types](#) (page 19-60)."

- 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 \(page 19-63\)"](#)
- ["Changing the Evaluation Point Attribute \(page 19-62\)"](#)
- ["About Resetting Authorization Scheme State \(page 19-63\)"](#)

19.5.2.3 Editing 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. To learn more, see ["About the Search Bar \(page 8-93\)"](#).

5. Select an authorization scheme.
6. Edit the appropriate attributes.
To learn more, see field-level Help. See ["Viewing Field-Level Help \(page 1-27\)"](#).
7. To save your changes, click **Apply Changes**.

19.5.2.4 Changing the Evaluation Point Attribute

You can 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 of 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.

19.5.2.5 About Resetting 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*

19.5.3 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](#) (page 19-64)
- [Attaching an Authorization Scheme to a Page](#) (page 19-64)
- [Attaching an Authorization Scheme to a Control or Component](#) (page 19-64)

19.5.3.1 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](#) (page 6-25)."
7. To define a new authorization scheme, click **Define Authorization Schemes**.

19.5.3.2 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.
4. To access the Edit Page:
 - Tree view - Under Page Rendering, double-click the page title at the top of the tree.
 - Component view- Under Page, click the page name.
5. Scroll down to Security and make a selection from the Authorization Scheme list.

19.5.3.3 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.
4. Click the name of the component or control to which you want to apply the authorization scheme.
5. Scroll down to Security and make a selection from the Authorization Scheme list.

19.5.4 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.

20

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.

Topics:

- [Understanding Application Translation and Globalization Support](#) (page 20-1)
- [Specifying the Primary Language for an Application](#) (page 20-5)
- [Using Format Masks for Items](#) (page 20-6)
- [About Translating Applications for Multibyte Languages](#) (page 20-7)
- [Understanding the Translation Process](#) (page 20-7)
- [Translating Messages](#) (page 20-15)
- [Translating Data That Supports List of Values](#) (page 20-58)
- [Understanding Supported Globalization Codes](#) (page 20-59)



See Also:

"Viewing Installed Translations" in *Oracle Application Express Administration Guide*

20.1 Understanding Application Translation and Globalization Support

You can develop applications in App Builder that can run concurrently in different languages. A single Oracle database instance and Oracle Application Express can support multiple database sessions customized to support different languages.

- [About the Translation Process](#) (page 20-1)
- [About Language Identification](#) (page 20-2)
- [About the Rules for Translating Database Applications](#) (page 20-2)
- [How Translated Applications Are Rendered](#) (page 20-2)
- [Making Application Attributes Translatable](#) (page 20-3)

20.1.1 About the Translation Process

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 \(page 20-7\)"](#)

20.1.2 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 \(page 20-5\)"](#)

20.1.3 About the 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`.

20.1.4 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 \(page 20-4\)"](#)
- ["Translating Data That Supports List of Values \(page 20-58\)"](#)

20.1.5 Making Application Attributes 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. This section describes best practices regarding how to make application attributes within your application translatable.

- [About Shortcuts that Support Translatable Messages \(page 20-3\)](#)
- [About Messages \(page 20-3\)](#)
- [About Dynamic Translation Text Strings \(page 20-4\)](#)
- [About Translating Region Titles \(page 20-4\)](#)
- [About Translating Templates \(page 20-5\)](#)

20.1.5.1 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 \(page 16-42\)"](#)

20.1.5.2 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 \(page 20-15\)"](#)

20.1.5.3 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 \(page 20-58\)"](#)

20.1.5.4 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 \(page 20-4\)](#)
- [Marking a Region as Not Translatable in Legacy Component View \(page 20-4\)](#)

20.1.5.4.1 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.
3. Expand the Advanced section.
4. For Exclude Title from Translation, select Yes.
5. Click **Save**.

20.1.5.4.2 Marking a Region as Not Translatable in Legacy Component View

To mark a region title as not translatable:

1. Navigate to the page in Component View. See "[Using Legacy Component View](#) (page 24-10)."
2. On the page, select the region title.
The Edit Region page appears.
3. Under Identification, select the **exclude title from translation** check box.

20.1.5.5 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](#) (page 11-32)"
- "[Substitutions](#) (page 6-23)"

20.2 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.

- 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 20-1](#) (page 20-6) describes available options:

Table 20-1 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.

- Configure other options as appropriate. To learn more about an attribute, see field-level Help.
- Click **Apply Changes**.

**See Also:**

"[Editing Application Attributes](#) (page 6-16)," "[Configuring Security Attributes](#) (page 6-23)," and "[Understanding Supported Globalization Codes](#) (page 20-59)"

20.3 Using Format Masks for Items

This section describes how to use format masks for items.

- [About Apply Format Masks to Items](#) (page 20-7)

- [Example of Editing Item Attributes](#) (page 20-7)

20.3.1 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.

20.3.2 Example of Editing Item Attributes

The following section describes how to edit item attributes for items having the source type of Database Column.

To edit item attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Navigate to the appropriate page.
4. Select an item.

The Edit Page Item page appears.

5. Under Name, make a selection from the Display As list.
6. Under Source, Format Mask, select or enter a format mask.



See Also:

["Viewing a Page in Legacy Component View](#) (page 24-10)"

20.4 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.

20.5 Understanding the Translation Process

This section describes the process of translating an application. 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.

- [Step 1: Map the Target Language](#) (page 20-8)
- [Step 2: Seed and Download to a Translation File](#) (page 20-9)
- [Step 3: Translate the XLIFF File](#) (page 20-11)
- [Step 4: Upload and Apply a Translated XLIFF Document](#) (page 20-12)
- [Step 5: Publish the Application](#) (page 20-14)
- [Manually Editing a Translation](#) (page 20-14)

 **See Also:**

- ["Translating Messages](#) (page 20-15)"
- ["Translating Data That Supports List of Values](#) (page 20-58)"

20.5.1 Step 1: Map the Target 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 **Map your primary language application to a translated application**.
The Mappings page appears.
3. Click **Create**.
4. On the Translation Application Mapping page:
 - **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 \(page 20-9\)"](#)

20.5.2 Step 2: Seed and Download to a Translation File

The second step is to seed the translation table and then export the translation text to a translation file.

- [Seeding Translatable Text \(page 20-9\)](#)
- [Downloading an XLIFF File \(page 20-9\)](#)

20.5.2.1 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.
```

20.5.2.2 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 \(page 20-10\)](#)
- [Downloading a XLIFF file for an Application \(page 20-10\)](#)
- [Downloading an XLIFF File for an Application Page \(page 20-10\)](#)
- [About Including XLIFF Target Elements \(page 20-11\)](#)

20.5.2.2.1 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

20.5.2.2.2 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 (page 20-9)"

20.5.2.2.3 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 then, 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 \(page 20-9\)"](#)

20.5.2.2.4 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.

20.5.3 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

20.5.4 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](#) (page 20-12)
- [Applying an Uploaded XLIFF Document](#) (page 20-12)
- [Deleting an Uploaded XLIFF Document](#) (page 20-13)

20.5.4.1 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.

20.5.4.2 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 \(page 20-5\)"](#)
- ["Step 5: Publish the Application \(page 20-14\)"](#)

20.5.4.3 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 \(page 20-5\)"](#)

20.5.5 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 \(page 20-5\)"](#)

20.5.6 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. Map the target 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. In Translate to Text, enter the appropriate text.
 - c. Click **Apply Changes**.
8. To access a Grid Edit page:
 - a. Click **Grid Edit**.
The Translatable Text - Grid Edit page appears.
 - b. In Translate To, enter the appropriate text.
 - c. Click **Apply Changes**.

 **See Also:**

- ["Step 1: Map the Target Language \(page 20-8\)"](#)
- ["Seeding Translatable Text \(page 20-9\)"](#)
- ["Customizing Interactive Reports in a Running Application \(page 8-92\)"](#)

20.6 Translating Messages

This section describes when and how to translate messages in an Oracle Application Express application.

- [About Translating Messages \(page 20-16\)](#)
- [Translating Messages Used in PL/SQL Procedures \(page 20-16\)](#)
- [Interactive Grid Messages Requiring Translation \(page 20-17\)](#)
- [Translating Messages Used Internally by Oracle Application Express \(page 20-35\)](#)

20.6.1 About Translating Messages

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.

20.6.2 Translating Messages Used in PL/SQL Procedures

This section explains how to translate messages used in PL/SQL procedures.

- [About Translating Messages Used in PL/SQL Procedures](#) (page 20-16)
- [Defining a Translation Message](#) (page 20-16)



See Also:

`APEX_LANG` in *Oracle Application Express API Reference*

20.6.2.1 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.

20.6.2.2 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**.

20.6.3 Interactive Grid Messages Requiring Translation

Interactive Grid Messages Requiring Translation lists the interactive grid messages that require translation.

Table 20-2 Interactive Grid Messages Requiring Translation

Message Name	English Text
<code>APEX.GV.BREAK_COLLAPSE</code>	Collapse control break
<code>APEX.GV.BREAK_EXPAND</code>	Expand control break
<code>APEX.GV.DELETED_COUNT</code>	%0 rows deleted
<code>APEX.GV.DUP_REC_ID</code>	Duplicate identity
<code>APEX.GV.FIRST_PAGE</code>	First
<code>APEX.GV.LAST_PAGE</code>	Last
<code>APEX.GV.LOAD_MORE</code>	Load More Rows
<code>APEX.GV.NEXT_PAGE</code>	Next
<code>APEX.GV.PAGE_RANGE_XY</code>	%0 - %1
<code>APEX.GV.PAGE_RANGE_XYZ</code>	%0 - %1 of %2
<code>APEX.GV.PREV_PAGE</code>	Previous
<code>APEX.GV.ROW_ADDED</code>	Added
<code>APEX.GV.ROW_CHANGED</code>	Changed
<code>APEX.GV.ROW_DELETED</code>	Deleted
<code>APEX.GV.SELECT_PAGE_N</code>	Page %0
<code>APEX.GV.SELECTION_COUNT</code>	%0 rows selected
<code>APEX.GV.SORT_ASCENDING</code>	Sort Ascending
<code>APEX.GV.SORT_ASCENDING_ORDER</code>	Sort Ascending %0
<code>APEX.GV.SORT_DESCENDING</code>	Sort Descending
<code>APEX.GV.SORT_DESCENDING_ORDER</code>	Sort Descending %0
<code>APEX.GV.SORT_OFF</code>	Don't Sort
<code>APEX.GV.TOTAL_PAGES</code>	Total %0
<code>APEX.IG.ACC_LABEL</code>	Interactive Grid %0
<code>APEX.IG.ACTIONS</code>	Actions
<code>APEX.IG.ADD</code>	Add
<code>APEX.IG.ADD_ROW</code>	Add Row

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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)
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

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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
</p> <p>The Aggregation list displays defined aggregations. Disable an existing aggregation by deselecting it.
</p> <p>Click Add (&plus;) to create a new aggregation, or Delete (&minus;) to remove an existing aggregation.</p></p> <p><p>Aggregation Settings
</p> <p>Use the form on the right to define the aggregation.
</p> <p>Select the Column name and Aggregation type.
</p> <p>Optionally, enter a tooltip for the aggregation.
</p> <p>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 20-2 (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 20-2 (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.COLUMNS	<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.COLUMNS_TITLE	Columns Help

Table 20-2 (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 20-2 (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 20-2 (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 20-2 (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>Sort List</p> The Sort dialog displays a list of configured sorting rules.
 Click Add (&plus;) to create a sort column, or Delete (&minus;) to remove a sort column.
 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>Sort Settings</p> Select a sort column, the sort direction, and how to order null columns (columns with no value).</p> <p> Note: Data can be sorted by columns which are not displayed; however, not all columns may be sortable.
 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 20-2 (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
APEX.IG.MIN	Minimum
APEX.IG.MIN_OVERALL	Overall Minimum

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
APEX.IG.OPEN_COLUMN	Open
APEX.IG.OPERATOR	Operator
APEX.IG.ORIENTATION	Orientation
APEX.IG.PIE	Pie

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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 with %0 rows and %1 columns. Report: %2, View: %3, Displayed Rows Start: %4, Displayed Rows End: %5
APEX.IG.TARGET_COLUMN	Target
APEX.IG.TEXT_COLOR	Text Color

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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

Table 20-2 (Cont.) Interactive Grid Messages Requiring Translation

Message Name	English Text
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
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

20.6.4 Translating Messages Used Internally by Oracle Application Express

Oracle Application Express is translated into German, Spanish, French, Italian, Japanese, Korean, Brazilian Portuguese, Simplified Chinese, and Traditional Chinese. 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.

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](#) (page 20-36)
- [Editing Text Messages](#) (page 20-36)
- [Internal Messages Requiring Translation](#) (page 20-37)
- [Interactive Report Messages Requiring Translation](#) (page 20-41)

20.6.4.1 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 \(page 20-37\)"](#)
- ["Interactive Report Messages Requiring Translation \(page 20-41\)"](#)

20.6.4.2 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.

20.6.4.3 Internal Messages Requiring Translation

Internal Messages Requiring Translation lists the internal messages that require translation.

Table 20-3 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

Table 20-3 (Cont.) Internal Messages Requiring Translation

Message Name	English Text
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.
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.

Table 20-3 (Cont.) Internal Messages Requiring Translation

Message Name	English Text
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
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_COL_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

Table 20-3 (Cont.) Internal Messages Requiring Translation

Message Name	English Text
LAYOUT.T_#EXPAND_COLLAPSE_NAV_LABEL#	Label for Expand / Collapse Navigation
LAYOUT.T_#EXPAND_COLLAPSE_SIDE_COL_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
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

Table 20-3 (Cont.) Internal Messages Requiring Translation

Message Name	English Text
WV_FLOW_CUSTOMIZE.T_THEME_STYLE	Appearance
WV_FLOW_DATA_EXPORT.AGG_COLUMN_IDX_NOT_EXIST	The column index referenced in the aggregate %0 does not exist.
WV_FLOW_DATA_EXPORT.COLUMN_BREAK_MUST_BE_IN_THE_BEGGINING	The column break needs to be in the beggining of the columns array.
WV_FLOW_DATA_EXPORT.COLUMN_GROUP_IDX_NOT_EXIST	The column group index referenced in %0 does not exist.
WV_FLOW_DATA_EXPORT.HIGHLIGHT_COLUMN_IDX_NOT_EXIST	The column index referenced in the highlight %0 does not exist.
WV_FLOW_DATA_EXPORT.PARENT_GROUP_IDX_NOT_EXIST	The parent group index referenced in %0 does not exist.
WV_FLOW_UTILITIES.CAL	Calendar
WV_FLOW_UTILITIES.CLOSE	Close
WV_FLOW_UTILITIES.OK	Ok
WV_FLOW_WEB_SERVICES.AUTHENTICATION_FAILED	Authentication failed.
WV_FLOW_WEB_SERVICES.NO_VALID_OAUTH_TOKEN	OAuth access token not available or expired.
WV_FLOW_WEB_SERVICES.UNSUPPORTED_OAUTH_TOKEN	Server responded with unsupported OAuth token type.
WV_RENDER_REPORT3.FOUND_BUT_NOT_DISPLAYED	Minimum row requested: %0, rows found but not displayed: %1
WV_RENDER_REPORT3.SORT_BY_THIS_COLUMN	Sort by this column.
WV_RENDER_REPORT3.UNSAVED_DATA	This form contains unsaved changes. Press Ok to proceed without saving your changes.
WV_RENDER_REPORT3.X_Y_OF_MORE_THAN_Z	row(s) %0 - %1 of more than %2
WV_RENDER_REPORT3.X_Y_OF_Z	row(s)%0 - %1 of %2
WV_RENDER_REPORT3.X_Y_OF_Z_2	%0 - %1 of %2

20.6.4.4 Interactive Report Messages Requiring Translation

[Table 20-4](#) (page 20-41) lists the interactive report messages that require translation.

Table 20-4 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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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
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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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
APEXIR_COLUMN_FILTER	Filter...
APEXIR_COLUMN_HEADING	Column Heading
APEXIR_COLUMN_HEADING_MENU	Column Heading Menu
APEXIR_COLUMN_INFO	Column Information
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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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.
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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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
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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
APEXIR_FUNCTION	Function
APEXIR_FUNCTION_N	Function %0
APEXIR_FUNCTIONS	Functions
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 20-4 (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 20-4 (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	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.
APEXIR_HELP_DETAIL_VIEW	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.
APEXIR_HELP_DOWNLOAD	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.

Table 20-4 (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	<p>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> <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	<p>Highlighting allows you to define a filter. The rows that meet the filter are highlighted using the characteristics associated with the filter.</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. </p>
APEXIR_HELP_PIVOT	<p>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> <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 20-4 (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 20-4 (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> <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>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 20-4 (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
APEXIR_MAX_QUERY_COST	The query is estimated to exceed the maximum allowed resources. Please modify your report settings and try again.

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
APEXIR_MAX_ROW_CNT	This query returns more than %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_MULTIIR_PAGE_REGION_STATIC_ID_REQUIRED	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.
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.

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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
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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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.
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 *

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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
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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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
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

Table 20-4 (Cont.) Interactive Report Messages Requiring Translation

Message Name	English Text
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

20.7 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](#) (page 20-58)



See Also:

`APEX_LANG` in *Oracle Application Express API Reference*

20.7.1 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**.

20.8 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](#) (page 20-59)



See Also:

["Specifying the Primary Language for an Application](#) (page 20-5)"

20.8.1 Oracle Application Express Globalization Codes

[Table 20-5](#) (page 20-59) describes the globalization codes in Oracle Application Express. `NLS_LANGUAGE` and `NLS_TERRITORY` determine the default presentation of numbers, dates, and currencies.

Table 20-5 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

Table 20-5 (Cont.) Oracle Application Express Globalization Codes

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
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
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

Table 20-5 (Cont.) Oracle Application Express Globalization Codes

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
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 (Luxemgourg)	de-lu	GERMAN	LUXEMBOURG
German (Switzerland)	de-ch	GERMAN	SWITZERLAND
Greek	el	GREEK	GREECE
Gujarati	gu	GUJARATI	INDIA
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

Table 20-5 (Cont.) Oracle Application Express Globalization Codes

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
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
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

Table 20-5 (Cont.) Oracle Application Express Globalization Codes

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
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

21

Managing Application Performance

This section describes how to improve application performance.

Topics:

- [About Best Practices to Improve Performance](#) (page 21-1)
- [Identifying Performance Issues](#) (page 21-2)
- [Utilizing Logs and Reports](#) (page 21-3)
- [About Utilizing Database Reporting](#) (page 21-6)
- [About Database Parameters that Impact Performance](#) (page 21-6)
- [About Limiting Resources](#) (page 21-7)
- [About Uploading Static Files to Your Web Server](#) (page 21-7)
- [Creating Custom Activity Reports Using APEX_ACTIVITY_LOG](#) (page 21-8)



See Also:

["Debugging an Application](#) (page 22-1)"

21.1 About Best Practices to Improve Performance

For applications having a large number of concurrent users, maintaining optimal performance is critical. To optimize your application's performance, remember to use the following features:

- 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](#) (page 2-24)"
- ["Understanding Substitution Strings](#) (page 2-26)"

21.2 Identifying Performance Issues

This section describes some key strategies to help identify and resolve performance issues with your Oracle Application Express application.

- [About Troubleshooting Slow Running Applications](#) (page 21-2)
- [Diagnosing Performance Issues](#) (page 21-2)
- [About Debugging Problematic SQL Queries](#) (page 21-3)
- [About Addressing Slow Queries](#) (page 21-3)

21.2.1 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

21.2.2 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.

21.2.3 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.

21.2.4 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.

See Also:

- ["Accessing Page Attributes in Legacy Component View \(page 24-12\)"](#)
- ["Duplicate Submission \(page 24-22\)"](#)
- ["About Using Collections \(page 17-7\)"](#)

21.3 Utilizing Logs and Reports

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 \(page 21-4\)](#)
- [Viewing the Application Dashboard Report \(page 21-4\)](#)
- [Workspace Activity Reports \(page 21-5\)](#)
- [Instance Activity Reports \(page 21-5\)](#)

 **See Also:**

- "APEX_DEBUG" in *Oracle Application Express API Reference*
- "Utilizing Debug Mode (page 22-4)"

21.3.1 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*

21.3.2 Viewing 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.

21.3.3 Workspace Activity Reports

Workspace administrators are users who perform administrator tasks specific to a workspace. 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*

21.3.4 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 packaged 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](#) (page 19-2)"

21.4 About 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](#) (page 22-3)"
- *Oracle Database Performance Tuning Guide*
- *Oracle Application Express SQL Workshop Guide*

21.5 About Database Parameters that Impact 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_050100` schema. Accurate statistics allow the cost-based optimizer to develop better query plans for accessing the application metadata.

21.6 About Limiting Resources

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*

21.7 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 \(page 16-12\)"](#)
- ["Managing Static Workspace Files \(page 16-17\)"](#)

21.8 Creating Custom Activity Reports Using APEX_ACTIVITY_LOG

This section describes how to use the use `APEX_ACTIVITY_LOG` view to query activity for the current workspace.

- [Enabling Logging for an Application \(page 21-8\)](#)
- [APEX_ACTIVITY_LOG \(page 21-9\)](#)
- [APEX_ACTIVITY_LOG Sample Query \(page 21-10\)](#)

21.8.1 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*

21.8.2 APEX_ACTIVITY_LOG

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 21-1](#) (page 21-9) describes the columns in the `APEX_ACTIVITY_LOG` view.

Table 21-1 Columns in APEX_ACTIVITY_LOG

Column	Type	Description
<code>time_stamp</code>	DATE	Date and time that activity was logged at the end of the page view.
<code>component_type</code>	VARCHAR2(255)	Reserved for future use.
<code>component_name</code>	VARCHAR2(255)	Reserved for future use.
<code>component_attribute</code>	VARCHAR2(4000)	Title of page.
<code>information</code>	VARCHAR2(4000)	Reserved for future use.
<code>elap</code>	NUMBER	Elapsed time of page view in seconds.
<code>num_rows</code>	NUMBER	Number of rows processed on page.
<code>userid</code>	VARCHAR2(255)	User ID performing page view.
<code>ip_address</code>	VARCHAR2(4000)	IP address of client.
<code>ir_report_id</code>	NUMBER	Interactive report ID
<code>ir_search</code>	VARCHAR2	Interactive report search criteria entered by users.
<code>user_agent</code>	VARCHAR2(4000)	web browser user agent of client.
<code>flow_id</code>	NUMBER	Application ID.
<code>step_id</code>	NUMBER	Page number.
<code>session_id</code>	NUMBER	Oracle Application Express session identifier.
<code>sqlerrm</code>	VARCHAR2(4000)	SQL Error message.
<code>sqlerrm_component_type</code>	VARCHAR2(255)	Reserved for future use.
<code>sqlerrm_component_name</code>	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.

21.8.3 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.

22

Debugging an Application

This section describes approaches how to debug an application.

Topics:

- [Running Advisor to Check Application Integrity](#) (page 22-1)
- [Reviewing Session State](#) (page 22-3)
- [About Monitoring Application and Page Resource Use](#) (page 22-3)
- [Enabling SQL Tracing and Using TKPROF](#) (page 22-3)
- [Utilizing Debug Mode](#) (page 22-4)
- [About Removing Controls and Components to Isolate a Problem](#) (page 22-8)



See Also:

- "APEX_DEBUG" in *Oracle Application Express API Reference*
- ["Managing Application Performance](#) (page 21-1)"

22.1 Running Advisor to Check Application Integrity

The Oracle Application Express Advisor (Advisor) enables you 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](#) (page 22-1)
- [Running Advisor on an Entire Application](#) (page 22-1)
- [Running Advisor on a Single Page](#) (page 22-2)

22.1.1 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.

22.1.2 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**.

22.1.3 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**.

22.2 Reviewing Session State

Many applications are based on data contained within application controls. For example, buttons can display conditionally based on a value stored in session state. You can view current session state for your application by clicking the **Session** link on the Runtime Developer Toolbar.

See Also:

- ["Runtime Developer Toolbar \(page 24-29\)"](#)
- ["Viewing Session State \(page 2-14\)"](#)
- ["Managing Session State Values \(page 2-20\)"](#)
- "Managing Session State and User Preferences" in *Oracle Application Express Administration Guide*

22.3 About Monitoring 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.

22.4 Enabling SQL Tracing and Using TKPROF

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.

- To view online Help, invoke TKPROF without arguments.

See Also:

"Understanding SQL Trace and TKPROF" in *Oracle Database SQL Tuning Guide* for information about using the TKPROF program

22.5 Utilizing Debug Mode

Debug mode enables you to track down unexpected application behavior. This section describes how to enable and disable debug mode, access debug mode using f?p syntax, and view debug reports.

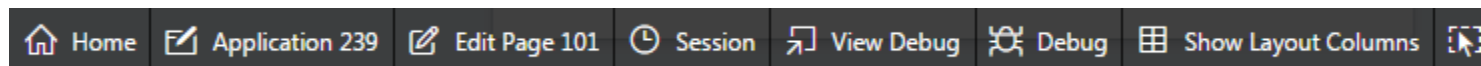
See Also:

"APEX_DEBUG" in *Oracle Application Express API Reference*

- [About Debug Mode](#) (page 22-4)
- [Enabling and Disabling Debug Mode for an Application](#) (page 22-5)
- [Running an Application in Debug Mode](#) (page 22-5)
- [Viewing Debug Reports in Development Mode](#) (page 22-6)
- [About Using f?p Syntax to Access Debug Mode](#) (page 22-6)
- [Viewing Debug Reports from a Running Application](#) (page 22-7)
- [Viewing Debug Reports](#) (page 22-7)

22.5.1 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 \(page 24-29\)"](#)
- ["APEX_DEBUG" in Oracle Application Express API Reference](#)

22.5.2 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 \(page 6-16\)"](#)

22.5.3 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 \(page 22-5\)"](#)

22.5.4 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.
6. To view steps that took the longest, hover over that step on the graph above the report.
The step details display.
7. 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 \(page 22-5\)"](#)

22.5.5 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 `LEVELn`, 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 \(page 2-17\)"](#)

22.5.6 Viewing Debug Reports from a Running Application

To view debug report from a running application:

1. Locate and run the application in Debug mode.
2. Click **View Debug** on the Runtime Developer Toolbar.
A list of Debug reports displays.
3. Click the View Identifier to the left of the session you want to view.
The Debug Message Data page 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.



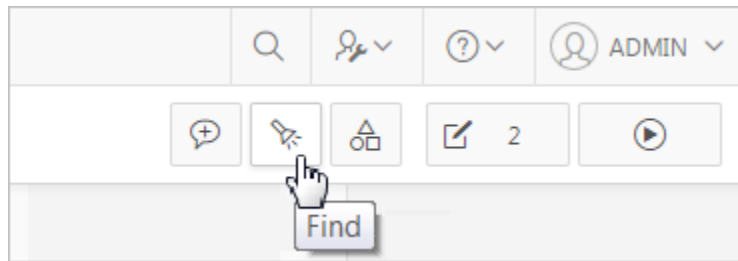
See Also:

- ["Running an Application in Debug Mode \(page 22-5\)"](#)
- ["Enabling and Disabling Debug Mode for an Application \(page 22-5\)"](#)

22.5.7 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 \(page 22-5\)"](#)

22.6 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 \(page 23-40\)"](#)

23

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*.

Topics:

- [About Suggested Environments](#) (page 23-2)
- [System Development Life Cycle Methodologies to Consider](#) (page 23-2)
- [Understanding the Deployment Process](#) (page 23-4)
- [How to Move an Application to Another Instance](#) (page 23-6)
- [How to Create a Custom Packaged Application](#) (page 23-8)
- [Exporting an Application and Application Components](#) (page 23-16)
- [Importing Export Files](#) (page 23-30)
- [Installing Export Files](#) (page 23-35)
- [About Publishing the Database Application URL](#) (page 23-38)
- [About Publishing the Worksheet Application URL](#) (page 23-39)
- [Using Build Options to Control Configuration](#) (page 23-40)

 **See Also:**

- ["Oracle Application Express Administration Guide"](#)
- ["Extending Application Capabilities \(page 18-1\)"](#)

23.1 About Suggested Environments

When developing applications in Oracle Application Express Oracle recommends you follow standard system development life cycle practices by having different environments for development, testing, and production. As a best practice, developers should only be allowed to 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 \(page 1-3\)"](#)

23.2 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. This section discusses several SDLC models that work well for developing applications in Oracle Application Express.

- [About Iterative Vs Planned Development \(page 23-2\)](#)
- [About the Advantages of Creating Prototypes \(page 23-3\)](#)
- [Waterfall \(page 23-3\)](#)
- [Spiral \(page 23-4\)](#)
- [Rapid Application Development \(page 23-4\)](#)

23.2.1 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.

23.2.2 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](#) (page 23-4)"
- "[Rapid Application Development](#) (page 23-4)"

23.2.3 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.

23.2.4 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.

23.2.5 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.

23.3 Understanding the Deployment Process

This section discusses different approaches to the deployment process.

- [About Moving an Application to Another Instance](#) (page 23-4)
- [Deployment Options to Consider](#) (page 23-5)
- [About Whether to Copy a Workspace](#) (page 23-5)
- [About Whether to Copy a Database](#) (page 23-5)
- [About the Application ID](#) (page 23-6)

23.3.1 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 \(page 23-6\)"](#)
- ["Using the Database Object Dependencies Report \(page 6-78\)"](#)
- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)

23.3.2 Deployment Options to Consider

When you develop an application in App Builder, 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.



See Also:

["Using the Database Object Dependencies Report \(page 6-78\)"](#)

23.3.3 About Whether to Copy a 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.

23.3.4 About Whether to Copy a 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.

23.3.5 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:

- ["Name \(page 6-17\)"](#) for information about defining an application alias
- ["Using Built-in Substitution Strings \(page 2-28\)"](#) for information about using `APP_ID` and `APP_ALIAS`

23.4 How to Move an Application to Another Instance

Whether you want to move an application to another workspace or just make a copy of it, deploying involves the following steps:

1. Move the supporting database objects (if appropriate). Review the Database Object Dependencies report to determine what objects to move.
2. Package the application definition with its supporting objects to create a custom packaged application.
3. Export the custom packaged application.
4. Import the exported files into the target Oracle Application Express instance.

You can import an application into your workspace regardless of the workspace in which it was developed.

Tip:

You can also move the application definition and all supporting objects manually.

- [About Managing Database Objects \(page 23-7\)](#)

 **See Also:**

- ["Using the Database Object Dependencies Report \(page 6-78\)"](#)
- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)
- ["Exporting an Application and Application Components \(page 23-16\)"](#)
- ["Importing Export Files \(page 23-30\)"](#)
- ["Deployment Options to Consider \(page 23-5\)"](#)
- ["Exporting an Application and Application Components \(page 23-16\)"](#)

23.4.1 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 \(page 6-78\)"](#)
- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)

23.5 How to Create a Custom Packaged Application

You can greatly simplify the steps needed to deploy an application by creating a custom packaged application on the Supporting Objects page.

- [How Creating a Custom Packaged Application Simplifies Deployment](#) (page 23-8)
- [Creating a Custom Packaged Application](#) (page 23-9)
- [Adding an Access Control List to a Custom Packaged Application](#) (page 23-12)
- [Installing Supporting Objects](#) (page 23-13)
- [Deleting Supporting Objects Scripts, Messages, and Installation Options](#) (page 23-13)
- [Upgrading a Custom Packaged Application](#) (page 23-14)
- [Deinstalling Supporting Objects](#) (page 23-15)
- [Viewing an Install Summary](#) (page 23-16)

See Also:

- ["How to Move an Application to Another Instance](#) (page 23-6)"
- ["Exporting an Application and Application Components](#) (page 23-16)"

23.5.1 How Creating a Custom Packaged 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 packaged 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, this feature has many 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 packaged 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 \(page 23-13\)"](#)
- ["Deinstalling Supporting Objects \(page 23-15\)"](#)
- ["Upgrading a Custom Packaged Application \(page 23-14\)"](#)

23.5.2 Creating a Custom Packaged Application

To create a custom packaged 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 Page \(page 23-9\)](#)
- [Supporting Objects Page \(page 23-9\)](#)

23.5.2.1 Accessing the Supporting Objects Page

You create a custom packaged application on the Supporting Objects page.

To access the Supporting Objects Page application:

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.

23.5.2.2 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.

See Also:

["Viewing Field-Level Help \(page 1-27\)"](#)

- [Installation](#) (page 23-10)
- [Upgrade](#) (page 23-11)
- [Deinstallation](#) (page 23-12)

23.5.2.2.1 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 packaged 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 packaged 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 \(page 6-23\)"](#)
- ["Using Build Options to Control Configuration \(page 23-40\)"](#)
- ["Exporting Build Options or Build Option Status \(page 23-42\)"](#)
- ["Understanding Cross-Site Scripting Protection \(page 19-21\)"](#)

23.5.2.2.2 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. See
- **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 Packaged Application \(page 23-14\)"](#)
- ["Understanding Cross-Site Scripting Protection \(page 19-21\)"](#)

23.5.2.2.3 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 \(page 19-21\)"](#)

23.5.3 Adding an Access Control List to a Custom Packaged 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 packaged 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
(page 19-39)"

23.5.4 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 packaged 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.

23.5.5 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 packaged 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.

23.5.6 Upgrading a Custom Packaged Application

You can define scripts to upgrade a previously published application on the Upgrade page.

- [Defining an Upgrade Script](#) (page 23-14)
- [Upgrading a Custom Packaged Application](#) (page 23-15)

23.5.6.1 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 packaged 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.

23.5.6.2 Upgrading a Custom Packaged 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 packaged application.

To upgrade a custom packaged 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 \(page 23-31\)"](#)

23.5.7 Deinstalling Supporting Objects

Once you create or install a custom packaged 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 packaged 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 Packaged Application \(page 23-9\)"](#)

23.5.8 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.

23.6 Exporting an Application and Application Components

Developers can export an application and related files to another Oracle Application Express instance.



Tip:

Developers 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 *Oracle Application Express Administration Guide*

- [About the Export Process \(page 23-17\)](#)
- [About The Export Page \(page 23-17\)](#)
- [Exporting an Application \(page 23-19\)](#)
- [Exporting Application Components \(page 23-21\)](#)
- [Exporting a Workspace \(page 23-23\)](#)
- [Exporting a Page in an Application \(page 23-24\)](#)
- [Exporting a Worksheet \(page 23-25\)](#)
- [About Exporting Static Files \(page 23-25\)](#)
- [Exporting Themes \(page 23-26\)](#)
- [Exporting Plug-ins \(page 23-26\)](#)

- [Exporting User Interface Defaults](#) (page 23-27)
- [Exporting Team Development Feedback](#) (page 23-29)
- [Exporting Script Files from the Script Repository](#) (page 23-30)

23.6.1 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](#) (page 23-19)"
- ["Exporting Application Components](#) (page 23-21)"
- ["Using Build Options to Control Configuration](#) (page 23-40)"

23.6.2 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.

The screenshot shows the Oracle App Builder interface. At the top, there are navigation tabs: ORACLE, App Builder (selected), SQL Workshop, Team Development, and Packaged Apps. Below these is an 'Export' button with an upward arrow icon. A secondary menu is open, listing: Export (selected), Workspace, Applications, Websheets, Themes, Plug-ins, User Interface Defaults, and Feedback. The main content area displays a grid of export options:

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 packaged application.

See Also:

- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)
- ["How to Move an Application to Another Instance \(page 23-6\)"](#)

23.6.3 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 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](#) (page 23-40)"
- "Changing Application Build Status Set During Deployment" in *Oracle Application Express Administration Guide*
- "[Importing Export Files](#) (page 23-30)"
- "Installing Exported Applications into a Runtime Environment" in *Oracle Application Express Administration Guide*

23.6.4 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:**

- ["Working with Shared Components \(page 16-1\)"](#)
- ["Importing an Application, Page or Component Export \(page 23-31\)"](#)
- ["Exporting Build Options or Build Option Status \(page 23-42\)"](#)

23.6.5 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*

23.6.6 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**.

 **See Also:**

- ["Viewing a Page in Page Designer \(page 7-22\)"](#)
- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["Exporting Application Components \(page 23-21\)"](#)
- ["Exporting an Application \(page 23-19\)"](#)
- ["Importing an Application, Page or Component Export \(page 23-31\)"](#)

23.6.7 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 \(page 5-1\)"](#)
- ["Importing a Websheet \(page 23-32\)"](#)

23.6.8 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 \(page 16-15\)"](#)
- ["Downloading All Static Workspace Files in a Zip \(page 16-19\)"](#)

23.6.9 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 \(page 23-33\)"](#)
- ["Using Themes \(page 11-1\)"](#)

23.6.10 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 \(page 23-33\)"](#)

23.6.11 Exporting User Interface Defaults

This section describes how to export User Interface Defaults.

- [About Exporting User Interface Defaults \(page 23-27\)](#)
- [Exporting User Interface Defaults from the Export Page \(page 23-28\)](#)
- [Exporting User Interface Defaults from the User Interface Defaults Page \(page 23-28\)](#)

23.6.11.1 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 \(page 23-34\)"](#)
- "Managing User Interface Defaults" in *Oracle Application Express SQL Workshop Guide*

23.6.11.2 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**.

23.6.11.3 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**.

23.6.12 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 \(page 23-35\)"](#)

23.6.13 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*

23.7 Importing Export Files

Once you export an application and any related files, you must import them into the target Oracle Application Express instance before you can 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 packaged application.

- [Importing an Application, Page or Component Export](#) (page 23-31)
- [About Importing Application Groups](#) (page 23-32)
- [Importing a Websheet](#) (page 23-32)
- [Importing Plug-ins](#) (page 23-33)
- [Importing Themes](#) (page 23-33)
- [Importing User Interface Defaults](#) (page 23-34)
- [Importing Team Development Feedback](#) (page 23-35)

 **See Also:**

- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)
- ["How to Move an Application to Another Instance \(page 23-6\)"](#)

23.7.1 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 Packaged Application](#) (page 23-8)"

23.7.2 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.

23.7.3 Importing a Websheet

To import a Websheet 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 **Websheet 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](#) (page 5-1)"

23.7.4 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.

23.7.5 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 \(page 11-1\)"](#)
- ["Exporting Themes \(page 23-26\)"](#)

23.7.6 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" \(page 23-27\)](#)

23.7.7 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**.

23.8 Installing Export Files

This section describes how to install export files.

- [About Installing Export Files](#) (page 23-35)
- [Accessing the Export Repository](#) (page 23-36)
- [Installing an Application Export from the Export Repository](#) (page 23-36)
- [Installing Other Files from the Export Repository](#) (page 23-37)
- [Deleting Files from the Export Repository](#) (page 23-38)
- [Installing Export Files from SQL*Plus](#) (page 23-38)
- [Using SQL*Plus to Install Export Files](#) (page 23-38)

23.8.1 About Installing Export Files

After you import an application and any related files into the target Oracle Application Express instance, the files are stored in the Export Repository. Next, you must install them.

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 packaged application.

**See Also:**

- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)
- "Managing Exported Files"
- "Installing Exported Applications into a Runtime Environment" in *Oracle Application Express Administration Guide*

23.8.2 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.

23.8.3 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 packaged application, the installer prompts you to install the custom packaged installation scripts. Follow the on-screen instructions.

 **See Also:**

- ["Accessing the Export Repository \(page 23-36\)."](#)
- ["Changing Application Build Status Set During Deployment" in *Oracle Application Express Administration Guide*](#)
- ["How to Create a Custom Packaged Application \(page 23-8\)"](#)

23.8.4 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](#) (page 23-36)."

23.8.5 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](#) (page 23-36)."

23.8.6 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.

23.8.7 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.

23.9 About Publishing the Database Application URL

Once you have deployed your application, loaded data, and created users, you can publish your production URL. You can determine the URL to 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 \(page 6-16\)"](#)

23.10 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.

23.11 Using Build Options to Control Configuration

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.

- [Creating Build Options](#) (page 23-41)
- [Managing Build Options](#) (page 23-41)
- [Selecting a Build Option](#) (page 23-41)
- [Exporting Build Options or Build Option Status](#) (page 23-42)
- [Viewing the Build Option Utilization Report](#) (page 23-43)

 **See Also:**

`GET_BUILD_OPTION_STATUS` and `SET_BUILD_OPTION_STATUS` procedures in "APEX_UTIL" in *Oracle Application Express API Reference*

23.11.1 Creating Build Options

You create a build option for an application on the Build Options 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 Security, click **Build Options**.
5. To create a new build option, click **Create**.
6. Follow the on-screen instructions.

23.11.2 Managing Build Options

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 Security, click **Build Options**.

The Build Options page appears.

5. Select a build option name.
The Create/Edit Build Option page appears.
6. For Status, select either **INCLUDE** or **EXCLUDE**.

23.11.3 Selecting a Build Option

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 attributes page. Most attributes pages contain a Configuration section where you can select defined build options.

See Also:

["Accessing Page Attributes in Legacy Component View \(page 24-12\)"](#)

23.11.4 Exporting Build Options or Build Option Status

You can export build options or build option status on the Component Export page. Exporting build option status is an effective way to toggle build options on or off within another environment. 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**.
The Component Export page appears.
2. 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 Packaged Application \(page 23-8\)"](#)
- ["Exporting Application Components \(page 23-21\)"](#)

23.11.5 Viewing the Build Option Utilization Report

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. Navigate to the Build Options page:
 - a. Navigate to the Workspace home page.
 - b. Click the **App Builder** icon.
 - c. Select an application.
 - d. Click **Shared Components**.
 - e. Under Security, click **Build Options**.
The Build Options page appears.
2. On the Build Options page, click **Utilization**.
The Build Option Utilization report appears.
3. Select a build option and click **Go**.

24

Managing Application Legacy Components

Legacy application components are components Oracle still supports but does not recommend using.

Topics:

- [Managing Pages in Legacy Component View](#) (page 24-1)
- [Managing Legacy Charts](#) (page 24-32)
- [Creating a Legacy Master Detail Form](#) (page 24-53)
- [Managing Legacy Tabular Forms](#) (page 24-56)
- [Managing Legacy Calendars](#) (page 24-61)
- [Viewing Legacy Components in Gallery](#) (page 24-68)

24.1 Managing Pages in Legacy Component View

Important:

Legacy Component View is disabled by default. To enable Legacy Component View, edit your user preferences and set **Enable Legacy Component View** to **Yes**. See "[Editing User Preferences](#) (page 1-23)".

Legacy Component View is a legacy user interface for editing application pages. Oracle recommends developers use Page Designer instead of Legacy Component View. Unlike Page Designer, Legacy Component View 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, you must run it. Also, some application components, such as interactive grids and Oracle JET based charts, cannot be edited or maintained using the Legacy Component View.

Tip:

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. However, when you click a component and the Property Editor highlights the corresponding attribute in the right pane. See "[Component View Tab](#) (page 7-14)".

- [Legacy Component View Window](#) (page 24-2)
- [Switching from Page Designer to Legacy Component View](#) (page 24-3)

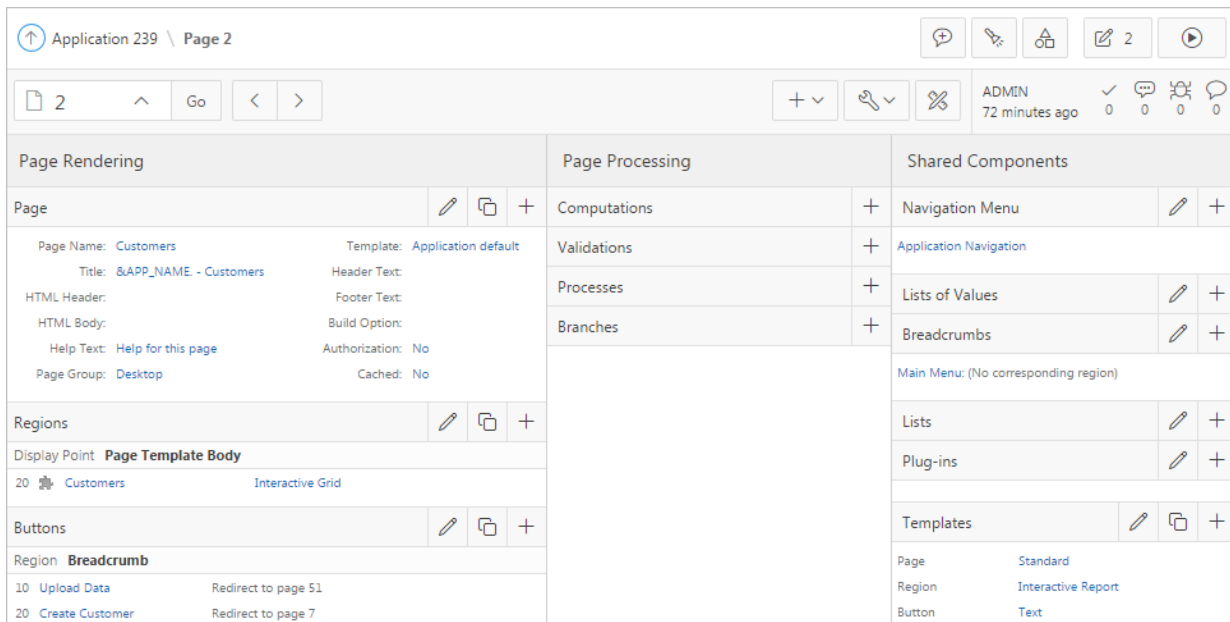
- [Understanding Legacy Component View UI Elements](#) (page 24-3)
- [Using Legacy Component View](#) (page 24-10)
- [Accessing Page Attributes in Legacy Component View](#) (page 24-12)
- [Running a Page in Legacy Component View](#) (page 24-28)
- [Runtime Developer Toolbar](#) (page 24-29)
- [Copying a Database Application Page in Legacy Component View](#) (page 24-30)
- [Deleting Pages in Legacy Component View](#) (page 24-31)

 **See Also:**

["Switching from Legacy Component View to Page Designer](#) (page 7-41)"

24.1.1 Legacy Component View Window

Legacy Component View is the traditional user interface for editing a page in App Builder. Component View groups user interface elements and application logic by component type. To edit a component, drill down on the component name or click the icons. Unlike Page Designer, Component View 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, you must run it.



Page Rendering	Page Processing	Shared Components
Page Page Name: Customers Title: &APP_NAME - Customers HTML Header: HTML Body: Help Text: Help for this page Page Group: Desktop Template: Application default Header Text: Footer Text: Build Option: Authorization: No Cached: No	Computations Validations Processes Branches	Navigation Menu Application Navigation Lists of Values Breadcrumbs Main Menu: (No corresponding region) Lists Plug-ins Templates Page: Standard Region: Interactive Report Button: Text
Regions Display Point: Page Template Body 20 Customers Interactive Grid		
Buttons Region: Breadcrumb 10 Upload Data Redirect to page 51 20 Create Customer Redirect to page 7		

The Legacy Component View window is divided into the 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.
- **Shared Components.** The Shared Components section lists common components that can be used by one or more pages within an application.

 **See Also:**

- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["Using Legacy Component View \(page 24-10\)"](#)

24.1.2 Switching from Page Designer to Legacy Component View

 **Important:**

Legacy Component View is disabled by default. To enable Legacy Component View, edit your user preferences and set **Enable Legacy Component View to Yes**. See ["Editing User Preferences \(page 1-23\)"](#).

To switch from Page Designer to Legacy Component View:

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 **Component View** tab in the central pane of Page Designer.
3. Click the **Legacy Component View** link beneath the Page Designer toolbar.

 **See Also:**

["Switching from Legacy Component View to Page Designer \(page 7-41\)"](#)

24.1.3 Understanding Legacy Component View UI Elements

Legacy Component View UI consists of the following UI elements:

 **Tip:**

Developer Navigation tools display in the header region on many Oracle Application Express pages. See "[Developer Navigation Tools](#) (page 1-6)."

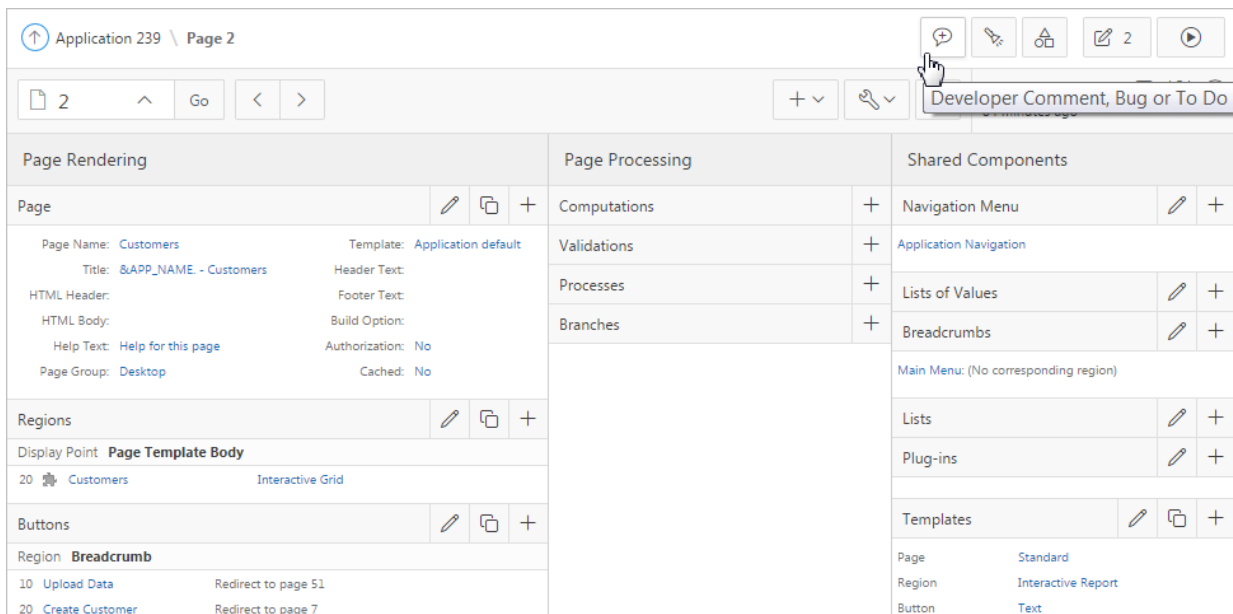
- [Component View Toolbar](#) (page 24-4)
- [Component View Navigation Bar](#) (page 24-5)
- [Page Rendering](#) (page 24-7)
- [Page Processing](#) (page 24-8)
- [Shared Components](#) (page 24-8)




























 **See Also:**

- "[Viewing a Page in Legacy Component View](#) (page 24-10)"
- "[Using Legacy Component View](#) (page 24-10)"

24.1.3.1 Component View Toolbar

The Component View toolbar displays at the top of the page. When you pass your cursor over an active button and menu, a tooltip displays as shown in the previous illustration.



Page Rendering	Page Processing	Shared Components
Page    Page Name: Customers Template: Application default Title: &APP_NAME. - Customers Header Text: HTML Header: Footer Text: HTML Body: Build Option: Help Text: Help for this page Authorization: No Page Group: Desktop Cached: No	Computations  Validations  Processes  Branches 	Navigation Menu   Application Navigation Lists of Values   Breadcrumbs   Main Menu: (No corresponding region) Lists   Plug-ins   Templates    Page Standard Region Interactive Report Button Text
Regions    Display Point Page Template Body 20  Customers Interactive Grid		
Buttons    Region Breadcrumb 10 Upload Data Redirect to page 51 20 Create Customer Redirect to page 7		

The Component View toolbar features the following buttons and menu options:

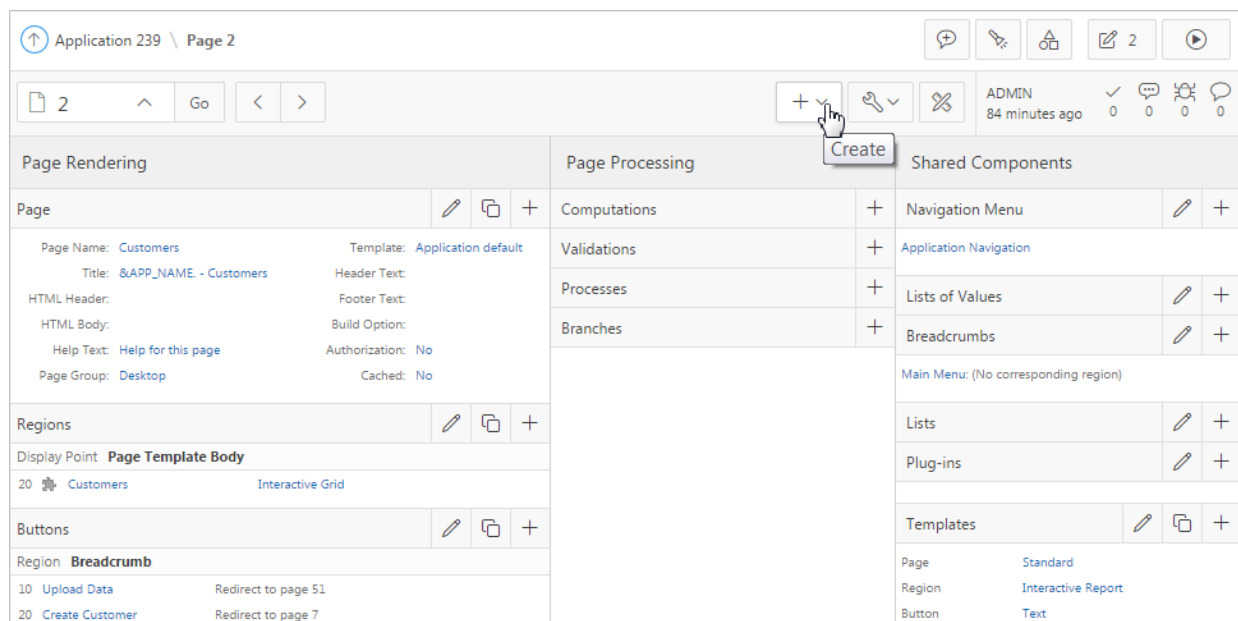
- **Developer Comment, Bug, or To Do.** Displays the Create a Developer comment, Bug or To Do dialog.
- **Find Icon.** The Find icon resembles a flashlight. Click the Find icon to search for items, pages, queries, tables, PL/SQL code, or images, view debug reports, view session state, and view errors.
- **Shared Components.** Links to the Shared Components page. Shared components can display or be applied on any page within an application.
- **Edit Page.** Displays the current page.
- **Run Page.** Runs the current page.




























 **See Also:**

- ["Adding Developer Comments \(page 6-68\)"](#)
- ["Managing Bugs \(page 4-16\)"](#)
- ["Tracking To Dos \(page 4-13\)"](#)
- ["Running a Page in Legacy Component View \(page 24-28\)"](#)

24.1.3.2 Component View Navigation Bar

A navigation bar displays at the top of the page beneath the breadcrumb menu. Use this bar to navigate between pages within an application.



Page Rendering	Page Processing	Shared Components
Page    Page Name: Customers Template: Application default Title: &APP_NAME - Customers Header Text: HTML Header: Footer Text: HTML Body: Build Option: Help Text: Help for this page Authorization: No Page Group: Desktop Cached: No	Computations  Validations  Processes  Branches 	Navigation Menu   Application Navigation Lists of Values   Breadcrumbs   Main Menu: (No corresponding region) Lists   Plug-ins   Templates    Page Standard Region Interactive Report Button Text
Regions    Display Point: Page Template Body 20  Customers Interactive Grid		
Buttons    Region: Breadcrumb 10 Upload Data Redirect to page 51 20 Create Customer Redirect to page 7		

- **Page Selector.** Navigates between pages. To navigate to page, enter a page number in the Page Selector and click **Go**.
- **Previous Page** - Click **Previous Page** (<) to view the previous application page.

- **Next Page** - Click **Next Page** (>) to view the next application page.
- **Create:**
 - **Page**. Access the Create Page Wizard.
 - **Page as Copy**. Access the Copy Page Wizard.
 - **Region**. Access to the Create Region Wizard.
 - **Page Component**. Access to the Create Page Component Wizard. Use this wizard to create items, buttons, branches, computations, processes, or validations.
 - **Shared Components**. Access to the Create Application Component Wizard. Shared components are common elements that can display or be applied on any page within an application.
 - **Page Group**. Links to the Page Group page. Use page groups to organize pages.
 - **Developer Comment**. Access the Developer Comments dialog. Developers can add comments to an application, a page, or a group of pages.
 - **Team Development**. Displays the submenus Feature, To Do, and Bug.
- **Utilities:**
 - **Delete Page**. Delete the current page.
 - **Advisor**. Access Oracle Application Express Advisor (Advisor). Use Advisor to check the integrity and quality of your Oracle Application Express application.
 - **Resequence Components**. Resequence page components, incrementing sequence numbers by 10.
 - **Lock**. Prevent conflicts during application development by locking pages in your application.
 - **Page Events**. Details all currently defined page controls and processes in the order in which the Application Express engine renders the page, invokes logic, and runs processes.
 - **Clean up**. Re-sequences the items on the page in increments of 10.
 - **Caching**. Links to the Caching page. Enabling caching is an effective way to improve the performance.
 - **Attribute Dictionary**. Access the Attribute Dictionary.
 - **History**. Displays a report of changes made to all element on the current page.
 - **Export**. Export the current page.
 - **Cross Page Utilities**. Access Cross Page Utilities.
 - **Application Utilities**. Access Application Utilities.
 - **Page Groups**. Links to the Page Group page. Use page groups to organize pages.
 - **Upgrade Application**. Upgrade an existing application.
- **Page Designer** - Click **Page Designer** to view the page in Page Designer.

**See Also:**

["Managing Pages in Page Designer \(page 7-1\)"](#)

24.1.3.3 Page Rendering

Page rendering is the process of generating a page from the database. Developers use the Page Rendering section of Component View to modify controls that impact the rendering of a page, including page attributes, regions, buttons, items, and page processes.

Table 24-1 Page Rendering Subsections

Subsection	Description
Page	<p>Page attributes control specific characteristics of a page such as the page name, display attributes such as the page title and the associated page template, header text, and the selected authorization scheme to name just a few.</p> <p>See Also: "Accessing Page Attributes in Legacy Component View (page 24-12)"</p>
Regions	<p>A region is an area on a page that serves as a container for content. Each page can have any number of regions. The content of a region is determined by the region source. For example, a region may contain a report based on a SQL query you define, or it may contain static HTML.</p> <p>You control the appearance of a region through a specific region template. You can use regions to group page controls (such as items or buttons). You can also create simple regions that do not generate additional HTML, or create elaborate regions that frame content within HTML tables or images.</p> <p>See Also: <i>Oracle Database Development Guide</i></p>
Buttons	<p>As you design your application, you can use buttons to direct users to a specific page or URL, or to enable users to submit a page. When you submit a page, the Application Express engine posts or processes information. A button implementation is determined by the button template. Buttons can be placed in predefined region template positions or among items in a form.</p> <p>See Also: "Creating Buttons (page 13-51)"</p>
Items	<p>Items are HTML form elements such as text fields, select lists, and check boxes with an associated session state. Item attributes affect the display and behavior of items on a page. For example, these attributes can impact where a label displays, how large an item is, and whether the item displays next to or below the previous item.</p> <p>There are two categories of items: page items and application items. Page-level items are placed on a page and have associated user interface properties, such as Display As, Label, and Label Template. Application-level items are not associated with a page and therefore have no user interface properties. An application item can be used as a global variable.</p> <p>See Also: "Understanding Page-Level Items (page 13-1)" and "Managing Application-Level Items (page 16-20)"</p>

Table 24-1 (Cont.) Page Rendering Subsections

Subsection	Description
Processes	Processes are logic controls used to execute Data Manipulation Language (DML) or PL/SQL. For example, you can use a process to populate session state at the time a page is rendered, to execute some type of logic (for example, using PL/SQL), or to make a call to the rendering engine. Typically a process performs an action. A process may be hand coded PL/SQL, or the invocation of a predefined process. See Also: " Understanding Page Processes (page 15-22)" and " Understanding Application Processes (page 15-32)"
Dynamic Actions	Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. You can specify an action that is performed based on a defined set of conditions. You can also specify which elements are affected by the action and when and how they are affected. See Also: " Managing Dynamic Actions (page 13-36)"

24.1.3.4 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 of Component View to specify application logic such as computations, validations, processes, and branches.

Table 24-2 Page Processing Subsections

Subsection	Description
Computations	Computations are units of logic used to assign session state to items and are executed at the time the page is processed.
Processes	Processes are logic controls used to execute Data Manipulation Language (DML) or PL/SQL. Processes are executed after the page is submitted.
Branches	Branches enable you to create logic controls that determine how the user navigates through the application.



See Also:

- "[Understanding Page Computations](#) (page 15-1)"
- "[Understanding Page Processes](#) (page 15-22)"
- "[Controlling Navigation Using Branches](#) (page 14-25)"
- "[About the When Button Pressed Attribute](#) (page 2-44)"

24.1.3.5 Shared Components

The Shared Components section contains common elements that can display or be applied on any page within an application.

Table 24-3 Shared Components Subsections

Subsection	Description
Tabs	<p>Tabs are an effective way to navigate between pages of an application. App Builder includes two types of tabs: standard tabs and parent tabs.</p> <p>An application having only one level of tabs uses a standard tab set. A standard tab set is associated with a specific page. You can use standard tabs to link users to other pages within your application. A parent tab set functions as a container to hold a group of standard tabs. Parent tabs give users another level of navigation and context (or sense of place) within the application.</p> <p>See Also: "Creating Tabs (page 14-2)"</p>
List of Values	<p>A list of values (LOV) is a static or dynamic definition used to display a specific type of page item, such as a radio group, check box, popup list, or select list. LOVs can be static (that is, based on a set of predefined display and return values) or dynamic (based on SQL queries that select values from tables). Once created, an LOV can then be referenced by one or more page items.</p> <p>You define LOVs at the application level by running the LOV Wizard and adding them to the List of Values repository.</p> <p>See Also: "Creating Lists of Values at the Application Level (page 16-24)"</p>
Breadcrumbs	<p>A breadcrumb is a hierarchical list of links that is rendered using a template. For example, you can display breadcrumbs as a list of links or as a breadcrumb path.</p> <p>See Also: "Creating Breadcrumbs (page 14-8)"</p>
Lists	<p>A list is a collection of links that is rendered using a template. For each list entry, you specify display text, a target URL, and other attributes that control when and how the list entry displays. You control the display of the list and the appearance of all list entries by linking the list to a template.</p> <p>See Also: "Creating Lists (page 16-27)"</p>
Theme	<p>A theme is a named collection of templates that defines the application user interface. Each theme contains templates for every type of application component and page control, including individual pages, regions, reports, lists, labels, menus, buttons, and list of values.</p> <p>See Also: "Using Themes (page 11-1)"</p>
Templates	<p>Templates control the look and feel of the pages in your application. As you create your application, you specify templates for pages, regions, reports, lists, labels, menus, buttons, and popup lists of values. Groups of templates are organized into named collections called themes.</p> <p>See Also: "Creating Custom Themes (page 11-32)"</p>
Security	<p>You can provide security for your application by specifying an authorization scheme. Authorization is a broad term for controlling access to resources based on user privileges.</p> <p>See Also: "Providing Security Through Authorization (page 19-59)"</p>
Navigation Bar	<p>Use a navigation bar to link users to various pages within an application. You can use a navigation bar to enable users to log in, log out, or link to Help text. The location of a navigation bar depends upon the associated page template. A navigation bar icon enables you to display a link from an image or text. When you create a navigation bar icon you can specify an image name, text, display sequence, and target location (a URL or page).</p> <p>See Also: "Creating a Navigation Bar Entry (page 14-19)"</p>

24.1.4 Using Legacy Component View

Developers can view and edit database application pages using Component View.

- [Viewing a Page in Legacy Component View](#) (page 24-10)
- [Edit All, Copy, and Create Icons](#) (page 24-11)
- [Creating a Page Control in Legacy Component View](#) (page 24-12)

24.1.4.1 Viewing a Page in Legacy Component View

Tip:

Legacy Component View is a legacy user interface for editing application pages. Oracle recommends developers use Page Designer instead of Legacy Component View. Legacy Component View 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, you must run it. Also, some application components, such as interactive grids and Oracle JET based charts, cannot be edited or maintained using the Legacy Component View.

To view a page in Legacy Component View:

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.
5. In the central pane, click 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 Component View tab presents user interface elements and application logic by component type. However, when you click a component, the Property Editor highlights the corresponding attribute in the right pane.

6. Click **Legacy Component View**.
Legacy Component View appears.

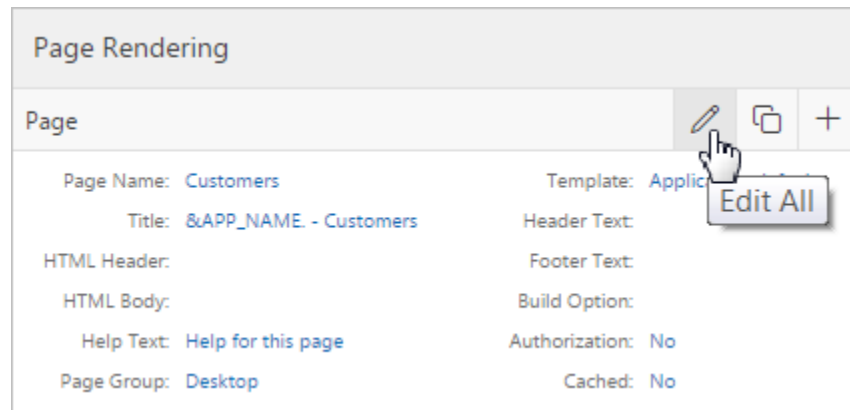
See Also:

- ["Managing Pages in Legacy Component View](#) (page 24-1)"
- ["Understanding Legacy Component View UI Elements](#) (page 24-3)"

24.1.4.2 Edit All, Copy, and Create Icons

Edit All Icon

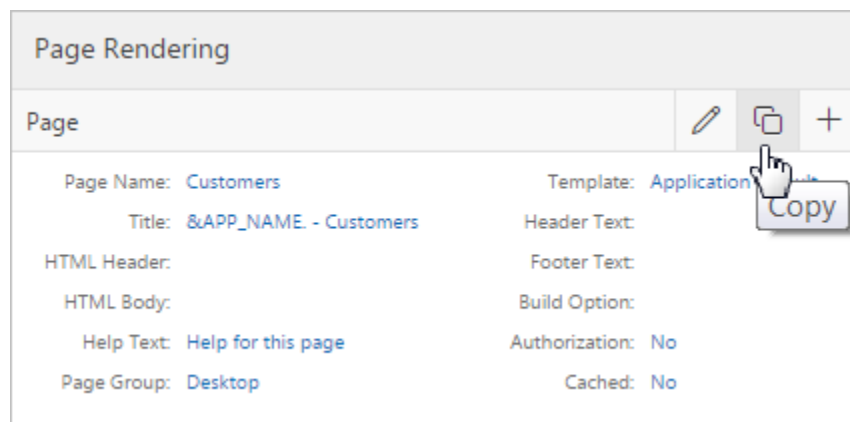
You can edit the controls, components, or logic within a given subsection by clicking the Edit All icon that displays to the right of the subsection title. The Edit All icon resembles a piece of paper with a pencil on top of it.



You can copy or create controls or components by clicking the Copy and Create icons.




Copy Icon

The Copy icon resembles two small overlapping pages. Click the Copy icon to make a copy of an existing control or component.



Create Icon

The Create icon resembles a plus (+) sign. Click the Create icon to create a control or component.

Page Rendering		Page Processing	
Page	  	Computations	+
Page Name: Customers	Template: Application default	Validations	+
Title: &APP_NAME. - Customers	Header Text:	Processes	+
HTML Header:	Footer Text:	Branches	+
HTML Body:	Build Option:		
Help Text: Help for this page	Authorization: No		
Page Group: Desktop	Cached: No		

24.1.4.3 Creating a Page Control in Legacy Component View

Page controls includes items, buttons, branches, computations, processes, or validations.

To add a page control in Component View:

1. View the page in Component View.
2. On the Navigation bar, click the **Create** menu and then select **Page Component**.
The Create Page Component wizard appears.
3. Select one of the following:
 - Item.
 - Button.
 - Branch.
 - Computation.
 - Process.
 - Validation.
4. Follow the on-screen instructions.

See Also:

- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.1.5 Accessing Page Attributes in Legacy Component View

Common page attributes you may want to edit include the page name or title, header and footer text, and the selected authorization scheme.

- [Editing Page Attributes in Component View \(page 24-13\)](#)
- [Page Attributes in Component View \(page 24-13\)](#)
- [Incorporating JavaScript Using Component View \(page 24-25\)](#)
- [Configuring Rejoin Sessions in Component View \(page 24-26\)](#)
- [Configuring Dialog Attributes in Component View \(page 24-27\)](#)

24.1.5.1 Editing Page Attributes in Component View

To edit page attributes:

1. View the page in Component View.
2. Under Page, click the page name or the **Edit All** icon.
The Page Attributes page appears.
3. Edit the appropriate attributes.
Required values are marked with a red asterisk (*).

Tip:

To learn more about any attribute, see field-level Help.

The Page Attribute page is divided into sections.

You can access these sections by scrolling down the page, or by clicking navigation buttons 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**.

4. To save your changes, click **Apply Changes**.

See Also:

- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["Viewing Field-level Help in Oracle Application Express \(page 1-27\)"](#)

24.1.5.2 Page Attributes in Component View

Component View Page Attributes page are divided into the following sections:

- [Name](#) (page 24-14)
- [Display Attributes](#) (page 24-14)
- [JavaScript](#) (page 24-15)
- [CSS](#) (page 24-16)
- [Dialog](#) (page 24-17)
- [HTML Header](#) (page 24-18)
- [Header and Footer](#) (page 24-18)
- [Read Only](#) (page 24-19)
- [Security](#) (page 24-19)
- [Duplicate Submission](#) (page 24-22)
- [Server Cache](#) (page 24-23)

- [Configuration](#) (page 24-23)
- [Error Handling](#) (page 24-24)
- [Help](#) (page 24-25)
- [Comments](#) (page 24-25)

24.1.5.2.1 Name

[Table 24-4](#) (page 24-14) describes general attributes for the current page such as the page name, an optional alphanumeric alias, and associated page groups.

Table 24-4 Page Attributes: Name

Attributes	Description
Name	Identifies the name of the current page for application developers. This name is used in numerous pages and reports, along with the page number and page title.
Page Alias	Enter an alphanumeric alias for this page. This alias must be unique within the current application. For example, if you were working on page 1 of application 100, you could create an alias called <code>home</code> . You could then access this page from other pages using the following <code>f?p</code> syntax: <code>f?p=100:home</code>
Group	Identify the page group you would like to associate with this page. Page groups do not affect functionality, but help developers manage the pages within an application. To remove a page from a group, select No Group Assigned and click Apply Changes . See Also: " Creating Page Groups (page 6-61)"

24.1.5.2.2 Display Attributes

[Table 24-5](#) (page 24-14) describes general display attributes for the current page such as the selected page template, standard tab set, title, and cursor focus.

Table 24-5 Page Attributes: Display Attributes

Attributes	Descriptions
Title	Enter a title to display in the title bar of the browser window. The Application Express engine uses the title you specify here in place of the <code>#TITLE#</code> substitution string used in the page template. This title is inserted between the HTML tags <code><TITLE></code> and <code></TITLE></code> .

Table 24-5 (Cont.) Page Attributes: Display Attributes

Attributes	Descriptions
Page Mode	<p>Select a page mode. Options include:</p> <ul style="list-style-type: none"> • Normal - The page displays as a normal Application Express application page. • 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. <p>See Also: "Creating Dialog Pages (page 6-49)"</p>
Page Template Options	<p>Template options enable you to select 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. To learn more about an attribute, see field-level Help.</p> <p>See Also: "Editing Themes (page 11-10)"</p>
Page CSS Classes	<p>Enter additional CSS classes as space a delimited list which are substituted if the page template contains the substitution string #PAGE_CSS_CLASSES#.</p> <p>See Also: "Editing Themes (page 11-10)"</p>
Cursor Focus	<p>Specify the cursor focus. Available options include:</p> <ul style="list-style-type: none"> • Select First item on page to have the cursor focus placed in the first field on the page. • Select Do not focus cursor if you do not want to include JavaScript. Select this option to prevent conflicts between generated JavaScript and custom JavaScript.
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. The page-level Media type overrides the application-level media type. The default value for this attribute is <code>NULL</code>. If both the page-level and application-level values for Media Type are <code>NULL</code>, the Media Type <code>text/html</code> is used.</p>

24.1.5.2.3 JavaScript

[Table 24-6](#) (page 24-16) describes attributes that enable you to include JavaScript on the current page.

Table 24-6 JavaScript Attributes

Attribute	Description
File URLs	<p>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.</p> <p>To view examples, see field-level Help.</p>
Function and Global Variable Declaration	<p>Enter JavaScript code (for example, functions or global variable declarations) for code to be used on this page. If the code is used on multiple pages, consider putting it into an external JavaScript file to avoid duplication.</p> <p>Code you enter here replaces the #PAGE_JAVASCRIPT# substitution string in the page template.</p> <p>Note: Do not to include opening or closing script tags. Just include the JavaScript code.</p> <p>To reference a shortcut, use:</p> <p>"SHORTCUTNAME"</p>
Execute when Page Loads	<p>Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle Application Express.</p> <p>Code you enter here replaces the #GENERATED_JAVASCRIPT# substitution string in the page template.</p> <p>Note: Do not to add any opening or closing script tags, just the JavaScript code.</p>

24.1.5.2.4 CSS

[Table 24-7](#) (page 24-16) describes attributes that enable you to include Cascading Style Sheet (CSS) files on the current page.

Table 24-7 CSS Attributes

Attribute	Description
File URLs	<p>Enter Cascading Style Sheet file URLs 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.</p> <p>File URLs you enter here replace the #PAGE_CSS# substitution string in the page template.</p> <p>Note: You do not need to include opening or closing link tags. Just include the file URL</p> <p>To view examples, see field-level Help.</p>

Table 24-7 (Cont.) CSS Attributes

Attribute	Description
Inline	<p>Enter Cascading Style Sheet code to be used on this page. If the code is used on multiple pages, consider putting it into an external file to avoid duplication and to leverage browser caching of static files.</p> <p>Code you enter here replaces the #PAGE_CSS# substitution string in the page template.</p> <p>Note: Do not include opening or closing style tags. Just include the cascading style sheet code.</p> <p>To view examples, see field-level Help.</p>

24.1.5.2.5 Dialog

Table 24-8 (page 24-17) describes attributes for defining a page which displays as a dialog. These attributes only apply when the Page Mode is set to **Modal Dialog** or **Non-Modal Dialog**.

 **Tip:**

These attributes overwrite similar settings defined at the page template-level. See "[Dialog](#) (page 11-63)".

Table 24-8 Page Attributes: Dialog

Attribute	Description
Width	<p>Enter the width of the dialog, in pixels or a percentage. This setting overwrites the default dialog width specified at page template-level.</p> <p>For a page using a Desktop user interface, the value should be set in pixels, for example, 500. For a page using the jQuery Mobile Smartphone user interface, the value should be set in percent, for example, 80%.</p> <p>The substitution string #DIALOG_WIDTH# in the dialog initialization code of the page template is replaced with the value entered in this item.</p>
Height	<p>Enter the height of the dialog in pixels, for example, 500.</p> <p>The substitution string #DIALOG_HEIGHT# in the dialog initialization code of the page template is replaced with value entered in this item.</p>
Maximum Width	<p>Enter the maximum width of the dialog, in pixels. This setting will overwrite the default dialog maximum width specified at page template-level.</p> <p>For a page using the jQuery Mobile Smartphone user interface, the maximum width is 500. The substitution string #DIALOG_MAX_WIDTH# in the dialog initialization code of the page template is replaced with the value entered in this item.</p>

Table 24-8 (Cont.) Page Attributes: Dialog

Attribute	Description
Attributes	<p>Dialog support in Oracle Application Express varies depending on the page user interface: Desktop user interfaces use a jQuery UI dialog and Mobile user interface uses the jQuery Mobile dialog widget. Oracle Application Express supports the following subset of dialog attributes: height, width, and maximum width.</p> <p>The substitution string <code>#DIALOG_ATTRIBUTES#</code> in the dialog initialization code defined at page template level, will be replaced by any value(s) entered in this item.</p> <p>To learn more, see field-level Help.</p>
CSS Classes	<p>Enter additional CSS classes you wish to be applied to the dialog, as a space delimited list. For example:</p> <pre>ui-dialog--wizard</pre> <p>Any occurrence of the substitution string <code>#DIALOG_CSS_CLASSES#</code> in the page template is replaced with the values specified in this field. This value overwrites any value set at page template-level.</p>
Chained Dialog	<p>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.</p> <p>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.</p>

24.1.5.2.6 HTML Header

[Table 24-9](#) (page 24-18) describes attributes for defining the page header and page HTML body text.

Table 24-9 HTML Header Attributes

Attribute	Description
HTML Header	<p>Enter text to replace the <code>#HEAD#</code> substitution string in the page template header. The values entered here are inserted after the HTML <code><HEAD></code> tag. Use this page attribute to add page specific meta tags.</p> <p>Note: JavaScript code or Cascading Style Sheets should be added to the dedicated JavaScript and CSS attributes.</p>
Page HTML Body Attribute	<p>Enter text to be substituted in for your page template's <code>#ONLOAD#</code> substitution string. To use this feature, your page template must include the <code>#ONLOAD#</code> substitution string.</p>

24.1.5.2.7 Header and Footer

[Table 24-10](#) (page 24-19) describes attributes for defining the header and footer text.

Table 24-10 Header Text and Footer Text

Attribute	Description
Header Text	Enter any text or HTML to appear immediately following the page header and before the body content.
Footer Text	Enter text or HTML to appear before the page template footer.

24.1.5.2.8 Read Only

Select a condition type from the list that must be met in order for this page to render page items as a read-only. Not selecting a condition causes the page items to render normally. The read only condition is evaluated after executing the processes with a processing point of Before Header.

 **Tip:**

If you want some page items (for example a search field) or all page items of a specific region to always render as enterable page items independent of how this condition evaluates, then set the Read Only condition of those regions or page items to **Never**.

24.1.5.2.9 Security

[Table 24-11](#) (page 24-19) describes attributes for defining authorization scheme and authentication requirements for the current page.

Table 24-11 Page Attributes: Security

Attribute	Description
Authorization Scheme	<p>Select an authorization scheme to be applied to the page. Authorization schemes are defined at the application-level and can be applied to many elements within the application.</p> <p>An authorization scheme is evaluated either once for each application session (at session creation), or once for each page view. If the selected authorization scheme evaluates to true, then the page displays and is subject to other defined conditions. If it evaluates to false, then the page does display and an error message displays.</p> <p>See Also: "Providing Security Through Authorization (page 19-59)"</p>
Authentication	<p>Specifies whether this page has been defined as public or requires authentication. If a page is identified as public, the page can be viewed before authentication. This attribute only applies to applications requiring authentication. The application's page sentry function can access this page attribute to identify pages that do not require prior authentication to view. The implementation of the authentication scheme's page sentry function determines if this attribute has any effect.</p> <p>See Also: "Establishing User Identity Through Authentication (page 19-45)"</p>

Table 24-11 (Cont.) Page Attributes: Security

Attribute	Description
Deep Linking	<p>Use this attribute to enable or prevent deep linking to an application page. 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. • Application Default - The application-level setting is used. <p>See Also: "Authentication (page 6-24)"</p>
Page Access Protection	<p>Note: This attribute only displays if Session State Protection has been enabled for the application. See "Understanding Session State Protection (page 19-26)."</p> <p>Select a Page Access Protection level for the current page:</p> <ul style="list-style-type: none"> • Unrestricted - The page may be requested using an 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.
Form Auto Complete	<p>Setting this attribute to On has no effect on the page. Setting this value to Off generates the following HTML in the FORM tag:</p> <pre>autocomplete="off"</pre>

Table 24-11 (Cont.) Page Attributes: Security

Attribute	Description
Browser Cache	<p>Note: This feature requires a modern browser that support the HTTP header response variable <code>cache-control</code>.</p> <p>Options include:</p> <ul style="list-style-type: none">• Application Default - The value at runtime is taken from the application's Browser Cache attribute.• Enabled - The browser saves the content of application's pages in its cache, both in memory and on disk. If a user presses the browser's back button, the page typically loads from the cache, not from the server.• Disabled - The browser does not save page content and requests content from the server. Oracle Application Express sends the HTTP header <code>cache-control: no-store</code> which instructs the browser to not cache the page content in memory and on disk. Disabling Browser Cache also prevents subtle back button issues with pages that use partial page refreshes (for example, pages with interactive reports). <p>From a security perspective and as a best practice, Oracle recommends the browser cache should be disabled. This prevents the browser from storing sensitive data and ensures the browser always request pages if the URL changes. Otherwise, it may even be possible to go back in the browser history after a logout and see cached content of the former session.</p>

Table 24-11 (Cont.) Page Attributes: Security

Attribute	Description
Rejoin Sessions	<p>The Rejoin Sessions attribute controls 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. A more restrictive instance-level setting overrides this page level value.</p> <p>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 (page 19-7)."</p> <p>From Rejoin Sessions, select an option:</p> <ul style="list-style-type: none"> • Application Default - Inherits the setting defined at application-level. See "Browser Security (page 6-30)." • 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. For applications with both public and authenticated pages, a session ID is defined after the end user authenticates. 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: <ul style="list-style-type: none"> 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. <p>See Also: "Browser Security (page 6-30)," "About Rejoin Sessions (page 19-7)," and "Configuring Rejoin Sessions" in <i>Oracle Application Express Administration Guide</i></p>

24.1.5.2.10 Duplicate Submission

Use the **Allow duplicate page submissions** list to specify whether users may process a page multiple times in a row. Set this attribute to **No** to prevent duplicate page submissions from being processed multiple times.

Examples of duplicate page submissions include:

- A user clicks the Submit button multiple times.
- You create a branch of type Branch to Page, and the user clicks the browser reload button.

In **On duplicate page submissions go to this URL**, enter a URL. Use this field if you set **Allow duplicate page submissions** to **No**. If you select No and a user attempts to

submit the same page twice, Oracle Application Express displays an error message and a link to the URL you specify.]

24.1.5.2.11 Server Cache

[Table 24-12](#) (page 24-23) describes attributes to enable caching for the current page.

Table 24-12 Page Attributes: Server Cache

Attribute	Description
Cache Page	Specify whether or not to enable page caching. Page caching can improve performance because the page displays from cache instead of displaying dynamically. Options include: <ul style="list-style-type: none"> • 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.
Cache Timeout	Displays if you select you select Enabled or Cached by User. Specify the amount of time for which a cached page remains valid.
Cache Page Condition	Specify a condition. If the condition returns true, the page is displayed from cache. If the condition returns false, the page is rendered dynamically and is not retrieved from cache.
Expression 1	Displays conditionally based on the selected Cache Page Condition. Enter values in this attribute based on the specific condition type selected.
Expression 2	Displays conditionally based on the selected Cache Page Condition. Enter values in this attribute based on the specific condition type selected.

See Also:

"Managing Cached Regions and Pages" in *Oracle Application Express Administration Guide* and "[Utilizing Region Caching in Component View](#) (page 12-22)"

24.1.5.2.12 Configuration

Select a build option for this page. You can use build options to enable or disable functionality. Most application attributes have a build option attribute. Do not specify a build option for the current page unless you plan to exclude the page in certain configurations.

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.

**See Also:**["Using Build Options to Control Configuration \(page 23-40\)"](#)

24.1.5.2.13 Error Handling

[Table 24-13](#) (page 24-24) describes attributes used to control or modify how an application logs errors.

In-line Error Notification Text, enter the error text that displays in the `#NOTIFICATION_MESSAGE#` template substitution string in the event an error occurs on the page.

Table 24-13 Page Attributes: Error Handling

Attribute	Description
In-line Error Notification Text	Enter the error text that displays in the <code>#NOTIFICATION_MESSAGE#</code> template substitution string in the event an error occurs on the page.
Error Handling Function	<p>Enter the name of a PL/SQL error function to modify the existing error message and display location 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 <code>#OWNER#</code> 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>See Also: <code>apex_error</code> in <i>Oracle Application Express API Reference</i></p> <p>Note: Error handling specified at the page-level overwrites any error handling function specified at the application-level. See "Editing Application Attributes (page 6-16)".</p>

**See Also:**["Page Templates \(page 11-59\)"](#)

24.1.5.2.14 Help

Use this attribute to enter Help text for the current page. Help text is displayed using a help system that you must develop.

See Also:

- ["Creating Help for Your Application \(page 10-33\)"](#)
- ["Using Shortcuts \(page 16-42\)"](#)

24.1.5.2.15 Comments

Use this attribute to record comments about the current page. These comments never display when the application is running.

24.1.5.3 Incorporating JavaScript Using Component View

You can include JavaScript into your application by adding it to the JavaScript attribute of the page. 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 the `.js` file.

In the following example, adding the code would make the `test` function accessible from anywhere on the current page.

To add JavaScript code to the JavaScript attribute:

1. View the page in Component View.
2. Under Page, click the **Edit All** icon.
3. Under JavaScript, enter code into the following attributes:

Tip:

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

- **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.
- **Function and Global Variable Declaration** - Enter JavaScript code (for example, functions or global variable declarations) for code to be used on this page. If the code is used on multiple pages, consider putting it into an external JavaScript file to avoid duplication.

- **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.

For example, adding the following would test a function accessible from anywhere on the current page.

```
function test(){  
    alert('This is a test.');
```

```
}
```

4. Click **Apply Changes**.

 **See Also:**

- "[Viewing a Page in Legacy Component View](#) (page 24-10)"
- "[Viewing Field-Level Help](#) (page 1-27)"

24.1.5.4 Configuring Rejoin Sessions in Component View

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](#) (page 19-7)."

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 Component View:

1. View the page in Component View.
2. Under Page, click the **Edit All** icon.
3. Scroll down to Security.
4. From Rejoin Sessions, select an option:
 - **Disabled** - All URLs must contain session IDs.
 - **Enabled for Public Sessions** - URLs to public pages do not have to contain session IDs if the user has not logged in yet.
 - **Enabled for All Sessions** - URLs do not have to contain session IDs if they are checksum protected or if they do not contain payload (a request parameter, clear cache or item data). This setting also requires that the application's Embed In Frames attribute is set to **Allow from same origin** or to **Deny**.

5. Click **Apply Changes**.

 **See Also:**

- ["About Rejoin Sessions \(page 19-7\)"](#)
- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.1.5.5 Configuring Dialog Attributes in Component View

 **Tip:**

Creating a dialog page is a multiple step process. To learn more, see ["Creating Dialog Pages \(page 6-49\)."](#)

To configure Dialog attributes in Component View:

1. View the page in Component View.
2. Under Page, click the **Edit All** icon.
3. Under Display Attributes, verify the Page Mode is set to either **Modal** or **Non-Modal**. See field-level Help for more information.
4. Scroll down to Dialog.
5. Under **Dialog**, and configure the attributes.

Table 24-14 Page Attributes: Dialog

Attribute	Description
Width	<p>Enter the width of the dialog, in pixels or a percentage. This setting overwrites the default dialog width specified at page template-level.</p> <p>For a page using a Desktop user interface, the value should be set in pixels, for example, 500. For a page using the jQuery Mobile Smartphone user interface, the value should be set in percent, for example, 80%.</p> <p>The substitution string <code>#DIALOG_WIDTH#</code> in the dialog initialization code of the page template is replaced with the value entered in this item.</p>
Height	<p>Enter the height of the dialog in pixels, for example, 500.</p> <p>The substitution string <code>#DIALOG_HEIGHT#</code> in the dialog initialization code of the page template is replaced with value entered in this item.</p>

Table 24-14 (Cont.) Page Attributes: Dialog

Attribute	Description
Maximum Width	<p>Enter the maximum width of the dialog, in pixels. This setting will overwrite the default dialog maximum width specified at page template-level.</p> <p>For a page using the jQuery Mobile Smartphone user interface, the maximum width is 500. The substitution string <code>#DIALOG_MAX_WIDTH#</code> in the dialog initialization code of the page template is replaced with the value entered in this item.</p>
Attributes	<p>Dialog support in Oracle Application Express varies depending on the page user interface: Desktop user interfaces use a jQuery UI dialog and Mobile user interface uses the jQuery Mobile dialog widget. Oracle Application Express supports the following subset of dialog attributes: height, width, and maximum width.</p> <p>The substitution string <code>#DIALOG_ATTRIBUTES#</code> in the dialog initialization code defined at page template-level, replaces any values entered in this item.</p> <p>To learn more about an attribute, see field-level Help.</p>
CSS Classes	<p>Enter additional CSS classes you wish to be applied to the dialog, as a space delimited list. For example:</p> <pre>ui-dialog--wizard</pre> <p>Any occurrence of the substitution string <code>#DIALOG_CSS_CLASSES#</code> in the page template is replaced with the values specified in this field. This value overwrites any value set at page template-level.</p>
Chained Dialog	<p>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.</p> <p>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.</p>

6. Click **Apply Changes**.

 **See Also:**

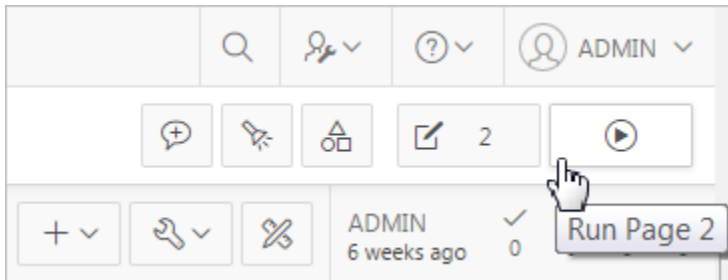
- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["Creating Dialog Pages \(page 6-49\)"](#)

24.1.6 Running a Page in Legacy Component View

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.

To run a page in Component View:

1. View the page in Component View.
2. To run the page, click the Run Page button in the upper right corner.



A rendered version of the application page appears.

See Also:

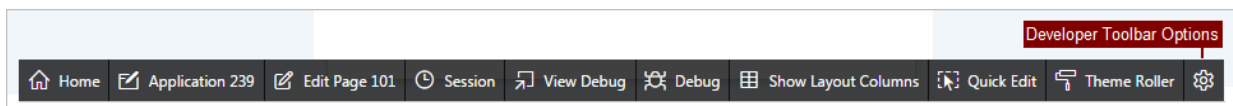
- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["Running an Application or Page \(page 2-7\)"](#)

24.1.7 Runtime Developer Toolbar

When you run a desktop application and view a page, the Runtime Developer toolbar displays at the bottom of any editable running page. Use the Developer toolbar to quickly edit the current application or currently running page, or view session state.

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 of the following controls:

- **Home** links to the Workspace home page. See "[Understanding the Workspace Home Page](#) (page 1-3)."
 - **Application** links to the Application home page. See "[Using the App Builder Home Page](#) (page 1-13)."
 - **Edit Page** displays the currently running page ID.
 - **Session** links you to session state information for the current page. See "[Using the Session Page](#) (page 2-15)."
 - **View Debug** displays the Debug reports. See "[Using the Session Page](#) (page 2-15)."
 - **Debug** toggles the page between **Debug** and **No Debug** mode. See "[Utilizing Debug Mode](#) (page 22-4)."
 - **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.
 - **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](#) (page 11-40)."
- 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](#) (page 11-3)."
- **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)

24.1.8 Copying a Database Application Page in Legacy Component View

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 in Component View.
2. Click **Create** and then **Page as Copy**.
3. For Copy Page Option, select one of the following:
 - **Page in this application**

- **Page in another application**
4. Follow the on-screen instructions.



See Also:

["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.1.9 Deleting Pages in Legacy Component View

Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries. Developers can only delete unlocked or pages you have locked.

- [Deleting a Page in Component View \(page 24-31\)](#)
- [Deleting a Page While Editing Page Attributes \(page 24-31\)](#)
- [Deleting Multiple Pages \(page 24-31\)](#)
- [Deleting a Range of Pages \(page 24-32\)](#)



See Also:

- ["Locking and Unlocking a Page \(page 6-64\)"](#)
- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.1.9.1 Deleting a Page in Component View

To delete a page in Component View:

1. View the page in Component View.
2. Verify the page number.
3. On the Component View toolbar, click **Utilities** and then **Delete Page**.
4. Click **Permanently Delete Page**.

24.1.9.2 Deleting a Page While Editing Page Attributes

To delete a page while editing page attributes:

1. Access the Page Attributes page.
2. Verify the page number and page name.
3. Click **Delete**.
4. Follow the on-screen instructions.

24.1.9.3 Deleting Multiple Pages

To delete multiple pages:

1. View the page in Component View.
2. Verify the page number.
3. On the Component View toolbar, click **Utilities, Cross Page**, and then **Delete Multiple Pages**.
4. Select the pages to be deleted and click **Delete Pages**.

24.1.9.4 Deleting a Range of Pages

To delete a range of pages:

1. View the page in Component View.
2. Verify the page number.
3. On the Component View toolbar, click **Utilities, Cross Page Utilities**, and then **Delete Multiple Pages**.
4. Click the **Delete Pages by Range** tab.
5. Enter the pages to be deleted and click **Delete Pages**.

24.2 Managing Legacy Charts

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](#) (page 10-23). "

Legacy charts in Oracle Application Express are based on the AnyChart HTML5 Chart Component and the AnyChart Flash Chart Component.

- [About Chart Types](#) (page 24-32)
- [Creating SQL Queries for Charts](#) (page 24-33)
- [Adding a Legacy Chart to a New Page](#) (page 24-35)
- [Managing Charts in Page Designer](#) (page 24-37)
- [Managing Charts in Legacy Component View](#) (page 24-41)
- [Understanding SVG Chart Support](#) (page 24-45)
- [Understanding HTML Chart Support](#) (page 24-51)
- [About Saving Flash Charts](#) (page 24-52)
- [Using Custom XML with Flash and HTML5 Charts](#) (page 24-52)

24.2.1 About Chart Types

HTML5 chart and Flash chart 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 HTML5 Charts](#) (page 24-33)
- [About Flash Charts](#) (page 24-33)

24.2.1.1 About 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>.

24.2.1.2 About 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.

24.2.2 Creating SQL Queries for Charts

This section describes the supported syntax for creating charts in App Builder.

- [Standard Chart Syntax](#) (page 24-33)
- [Dial Chart Syntax \(Flash and HTML5\)](#) (page 24-34)
- [Multiple Series Syntax \(Flash and HTML5\)](#) (page 24-34)
- [Range Chart Syntax \(Flash and HTML5\)](#) (page 24-34)
- [Scatter Chart Syntax \(Flash and HTML5\)](#) (page 24-34)
- [Candlestick Chart Syntax \(Flash and HTML5\)](#) (page 24-35)
- [Gantt Chart Syntax \(Flash Only\)](#) (page 24-35)

24.2.2.1 Standard Chart Syntax

You define a chart in App Builder using a wizard. For most chart wizards, 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
```

24.2.2.2 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
```

24.2.2.3 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   ...
```

24.2.2.4 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   ...
```

24.2.2.5 Scatter Chart Syntax (Flash and HTML5)

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   ...
```

24.2.2.6 Candlestick Chart Syntax (Flash and HTML5)

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 ...
```

24.2.2.7 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
```

24.2.3 Adding a Legacy Chart to a New Page

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.
 - b. Select a Page Type - Select **Legacy Page** and then select **AnyChart Chart**.
5. For Chart Type:
 - a. Chart Rendering - Select a chart type.
 - 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
 - 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. Query 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 Types \(page 24-32\)"](#)
- ["Creating SQL Queries for Charts \(page 24-33\)"](#)
- ["Viewing Field-Level Help \(page 1-27\)"](#)

24.2.4 Managing Charts in Page Designer

This section describes how to create and edit charts in Page Designer.

- [Adding a Chart in Page Designer \(page 24-37\)](#)
- [Editing Chart Attributes in Page Designer \(page 24-38\)](#)
- [Switching an HTML5 Chart to a Flash Chart \(page 24-39\)](#)
- [Enabling Asynchronous Updates \(page 24-39\)](#)

24.2.4.1 Adding a Chart in Page Designer

To add a chart to an existing page in Page Designer:

1. View the page to contain the item in Page Designer.
2. From 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 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. In the Rendering tab, locate the chart and select the **Attributes** node.
The Property Editor displays the Attributes for the chart.
 - b. Edit the attributes.
5. Edit the Series:
 - a. In the Rendering tab, select the **Series** node.
The Property Editor displays the Series attributes.
 - b. Edit the attributes.
 6. Click **Save**.



See Also:

"[Viewing a Page in Page Designer](#) (page 7-22)."

24.2.4.2 Editing Chart Attributes in Page Designer

Once you have created a chart, you can alter its display by editing chart attributes in Page Designer.



Note:

The deprecated chart type SVG cannot be edited using Page Designer.

To edit chart attributes:

1. View the page to contain the chart in Page Designer.
2. In the Rendering tab, locate the chart and select the **Attributes** node.
3. To view all attributes in the Property Editor, click **Show All** .
4. To search for a group or an attribute, enter a keyword in the Filter Properties field.

5. Edit the chart 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**.

 **Note:**

["Viewing a Page in Page Designer \(page 7-22\)"](#)

24.2.4.3 Switching an HTML5 Chart to a Flash Chart

Once you create an HTML5 chart, you can switch its rendering to Flash by editing chart attributes on the Chart Attributes page.

To switch an HTML5 chart to a Flash Chart:

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. Expand the Appearance section.
4. For Appearance, Rendering, select **Flash Chart**.
To learn more about an attribute, see field-level Help.
5. Click **Save**.

 **See Also:**

- ["Viewing a Page in Page Designer \(page 7-22\)"](#)
- ["Viewing Field-Level Help \(page 1-27\)"](#)

24.2.4.4 Enabling Asynchronous Updates

SVG, 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 \(page 24-40\)](#)
- [Creating a Refresh Dynamic Action in Page Designer \(page 24-40\)](#)

24.2.4.4.1 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 to contain the item in Page Designer.
2. In the Rendering tab, locate the chart and select the **Attributes** node.
The Property Editor displays **Attributes**.
3. To view all attributes in the Property Editor, click **Show All**.
4. To search for a group or an attribute, enter a keyword in the Filter Properties field.
5. Find **Automatic Refresh** and select **Yes**.
6. Find **Interval** and specify the refresh interval, in seconds.
7. Click **Save**.

**Note:**

"[Viewing a Page in Page Designer](#) (page 7-22)"

24.2.4.4.2 Creating a Refresh Dynamic Action in Page Designer

To create a Refresh dynamic action:

1. View the page that contain the item 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 Property Editor displays Dynamic Action attribute.

**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. To view all attributes in the Property Editor, click **Show All**.
5. 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.

6. In the Rendering tab, select the first action under True. In the Property Editor, edit the following Action attributes:
 - a. Action - Select **Refresh**.
 - b. Selection Type - Select **Region**.
 - c. Region - Select the region containing the chart.
7. Click **Save**.

 **Note:**

["Viewing a Page in Page Designer \(page 7-22\)"](#)

24.2.5 Managing Charts in Legacy Component View

Developers can create and manage charts in Component View.

- [Adding a Chart in Component View \(page 24-41\)](#)
- [Switching an HTML5 Chart to a Flash Chart \(page 24-43\)](#)
- [Editing Chart Attributes in Component View \(page 24-43\)](#)
- [Enabling Asynchronous Updates \(page 24-44\)](#)

 **See Also:**

[Managing Pages in Legacy Component View \(page 24-1\)](#)

24.2.5.1 Adding a Chart in Component View

To add a chart to an existing page in Component View:

1. Navigate to the page.
2. Create a new region. Under Region, click the **Create** icon.
The Create Region Wizard appears.
3. On Create Region, select **Chart** and click **Next**.
4. For Chart Type:

- a. Chart Rendering - Select a chart type.
 - b. Chart Type - Specify a chart type. Depending upon the chart type, you may be prompted to make additional selections.
 - c. Click **Next**.
5. For Display Attributes:
- a. Title - Enter a title for the region. This title displays if the region template you choose displays the region title.
 - b. Region Template - Enter a title for the region. This title displays if the region template you choose displays the region title.
 - c. Parent Region - Defines the parent region to which the new region belongs to.
 - d. Display Point - Identify a display point for this region. To learn more, see field-level Help.
To learn more, see field-level Help.
 - e. Click **Next**.
6. For Chart Attributes, specify the appropriate attributes and click **Next**.
To learn more about an attribute, see field-level Help.
7. For Source:
- a. 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.
 - b. Specify relevant chart attributes.
To learn more about an attribute, see field-level Help.
8. For Conditional Display:
- a. Condition Type - Select a condition type from the list that must be met in order for this component to be rendered processed.
 - b. Authorization Scheme - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.
 - c. Click **Create Region**.

 **See Also:**

- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["About Chart Types \(page 24-32\)"](#)
- ["Viewing Field-Level Help \(page 1-27\)"](#)
- ["Creating SQL Queries for Charts \(page 24-33\)"](#)
- ["Understanding Conditional Rendering and Processing \(page 2-10\)"](#)
- ["Providing Security Through Authorization \(page 19-59\)"](#)

24.2.5.2 Switching an HTML5 Chart to a Flash Chart

Once you create an HTML5 chart, you can switch its rendering to Flash by editing chart attributes on the Chart Attributes page.

To switch an HTML5 chart to a Flash Chart:

1. Navigate to the page.
2. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Chart Attributes**.
3. For Chart Type:
 - a. Chart Rendering - Select **Flash Chart**.
 - b. Chart Type - Specify a chart type. Note that 3D chart types are only supported for Flash charts, so you may want to change the chart type to use one of the 3D options.

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

4. Click **Apply Changes**.



See Also:

- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["Viewing Field-Level Help \(page 1-27\)"](#)

24.2.5.3 Editing Chart Attributes in Component View

Once you have created a chart, you can alter its display by editing chart attributes on the Chart Attributes page.

To access the Chart Attributes page:

1. Navigate to the page.
2. Access the Chart Attributes page. Under Region, select the region name.
3. Edit the appropriate attributes.
4. To learn more about an attribute, see field-level Help.
5. Click **Apply Changes**.



See Also:

- ["Viewing a Page in Legacy Component View \(page 24-10\)."](#)
- ["Viewing Field-Level Help \(page 1-27\)."](#)

24.2.5.4 Enabling Asynchronous Updates

SVG, 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](#) (page 24-44)
- [Creating a Refresh Dynamic Action](#) (page 24-44)

24.2.5.4.1 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. Navigate to the Chart Attributes page.
The Chart Attributes page appears.
2. Scroll down to Refresh.
3. From Asynchronous Update, select **Yes**.
4. In Update Interval (Seconds), enter the interval in seconds between chart updates. For optimal performance, select an interval that is greater than 2 seconds.



Note:

"[Editing Chart Attributes in Component View](#) (page 24-43)."

24.2.5.4.2 Creating a Refresh Dynamic Action

To create a Refresh dynamic action:

1. Navigate to the page containing the chart.
2. Locate the Dynamic Action section and click **Create**.
3. For Identification:
 - a. Name - Enter a name of the dynamic action.
 - b. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
 - c. Click **Next**.
4. For When:
 - a. Event - Specify the event to trigger the firing of the dynamic action.
 - b. Selection Type - Specify the selection type of the page elements that define when the dynamic action fires. See field-level Help to view examples for specific page items.
 - c. Click **Next**.
5. For True Action:

- a. Action - Select **Refresh**.
 - b. Click **Next**.
6. For Affected Elements:
 - a. Selection Type - Select **Region**.
 - b. Region - Select the region containing the chart.
 - c. Click **Next**.
7. Click **Create Dynamic Action**.

**Note:**

"[Viewing a Page in Legacy Component View](#) (page 24-10)"

24.2.6 Understanding SVG Chart Support

Previous releases of Oracle Application Express supported the creation of SVG charts. Although existing SVG charts are still supported, Oracle Application Express no longer supports the creation of new SVG charts.

**Tip:**

You can only edit SVG charts in Component View.

- [Upgrading SVG Charts to HTML5](#) (page 24-45)
- [Understanding SVG Chart Cascading Style Sheet Classes](#) (page 24-46)
- [Referencing a Custom SVG Chart Cascading Style Sheet](#) (page 24-49)
- [Specifying Custom CSS Styles Inline for SVG Charts](#) (page 24-50)
- [Displaying SVG Charts in Other Languages](#) (page 24-50)

24.2.6.1 Upgrading SVG Charts to HTML5

This section describes restrictions that apply when upgrading SVG charts to HTML5 and also describes the instructions for upgrading.

You can automatically upgrade single or multiple SVG charts to HTML5.

- [About SVG Chart Upgrade Restrictions](#) (page 24-45)
- [Upgrading an SVG Chart to HTML5](#) (page 24-46)
- [Upgrading all SVG Charts in an Application to HTML5](#) (page 24-46)

24.2.6.1.1 About SVG Chart Upgrade Restrictions

SVG charts can be upgraded with the following restrictions:

- Only number formats defined in axis format strings are migrated. Date and time formats are ignored.

- Number format elements containing the following are migrated:
0,9,D,G,, (comma),. (period),\$,C,L,FM
- The label for each series in the HTML5 chart is derived from each series' column alias. This differs from SVG charts, where the label for each series is derived from the Series Name attribute.
- In SVG charts, only the labels for the first series are used for the x-axis. In HTML5 charts, this has been enhanced so that all data appears, even if the data's label does not occur in the first series.

24.2.6.1.2 Upgrading an SVG Chart to HTML5

To upgrade an SVG chart to HTML5:

1. Navigate to the page.
2. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Chart Attributes**.
3. From the Tasks list, click **Upgrade to New HTML5 Chart**.
4. Click **Upgrade**.



Note:

["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.2.6.1.3 Upgrading all SVG Charts in an Application to HTML5

To upgrade all SVG Charts to HTML5 charts:

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. Click **Utilities** and click **Upgrade Application**.
The Upgrade Application Summary report appears.
4. For Upgrade SVG Charts to HTML5 Charts, click the number of candidate objects.
5. Select the objects to upgrade and click **Upgrade**.

24.2.6.2 Understanding SVG Chart Cascading Style Sheet Classes

When you create a SVG chart, Oracle Application Express renders it based on cascading style sheet (CSS) classes associated with the current theme. You can change the appearance of a chart by referencing another CSS or by overriding individual classes in the CSS section of the Edit Attributes page

The following sample contains the CSS classes for the dial chart in *Sample Application*. This example contains all the available CSS classes. Class names appear in boldface.

```
text{font-family:Verdana, Geneva, Arial, Helvetica, sans-serif;fill:#000000;}
tspan{font-family:Verdana, Geneva, Arial, Helvetica, sans-serif;fill:#000000;}
text.title{font-weight:bold;font-size:14;fill:#000000;}
text.moredatafound{font-size:12;}
rect.legend{fill:#EEEEEE;stroke:#000000;stroke-width:1;}
text.legend{font-size:10;}
#background{fill:#FFFFFF;stroke:none;}
rect.chartholderbackground{fill:#ffffff;stroke:#000000;stroke-width:1;}
#timestamp{text-anchor:start;font-size:9;}
text.tic{stroke:none;fill:#000000;font-size:12}
line.tic{stroke:#000000;stroke-width:1px;fill:none;}
#dial{stroke:#336699;stroke-width:2px;fill:#336699;fill-opacity:.5;}
#dial.alert{fill:#FF0000;fill-opacity:.5;}
#dialbackground{stroke:#000000;stroke-width:none;fill:none;filter:url(#MyFilter);}
#dialcenter{stroke:none;fill:#111111;filter:url(#MyFilter);}
#dialbackground-border{stroke:#DDDDDD;stroke-width:2px;fill:none;filter:url
(#MyFilter);}#low{stroke-width:3;stroke:#336699;}
#high{stroke-width:3;stroke:#FF0000;}
#XAxisTitle{letter-spacing:2;kerning:auto;font-size:14;fill:#000000;text-
anchor:middle;}
#YAxisTitle{letter-spacing:2;kerning:auto;font-size:14;fill:#000000;text-
anchor:middle;writing-mode:tb;}
.XAxisValue{font-size:8;fill:#000000;}
.YAxisValue{font-size:8;fill:#000000;text-anchor:end;}
.nodatafound{stroke:#000000;stroke-width:1;font-size:12;}
.AxisLine{stroke:#000000;stroke-width:2;fill:#FFFFFF;}
.GridLine{stroke:#000000;stroke-width:0.3;stroke-dasharray:2,4;fill:none;}
g.dataholder rect{stroke:#000000;stroke-width:0.5;}
.legenditem rect{stroke:#000000;stroke-width:0.5;}
```

Table 24-15 (page 24-47) describes all supported CSS classes. Note that certain classes only apply to specific chart types.

Table 24-15 Available SVG Chart CSS Classes

Class	Description
text	Defines the appearance of text that displays in a chart.
tspan	Defines the appearance of text that displays in a chart. <code>tspan</code> should match the definition of <code>text</code> .
text.title	Overrides the default chart text. Use this class for title text.
text.moredatafound	Defines the appearance of more datafound text.
rect.legend	Creates the rectangular box that holds the chart legend. To remove the legend border, change <code>rect.legend</code> to the following: <code>rect.legend{fill:#CCCC99;stroke:none;}</code>
text.legend	Defines the text that appears in the chart legend.

Table 24-15 (Cont.) Available SVG Chart CSS Classes

Class	Description
#background	Creates the entire background for the SVG plug-in. For a solid white background with no border, change #background to the following: <code>#background{fill:#FFFFFF;stroke:#FFFFFF;stroke-width:2;}</code>
rect.chartholderbackground	Not applicable to pie and dial charts. Creates the background of the rectangle that holds the chart data. For a clear background, change rect.chartholderbackground to the following: <code>rect.chartholderbackground(display:none;)</code>
#timestamp	Only applicable if the Asynchronous Update chart attribute is set to Yes. Controls the appearance of the update timestamp test. To disable the display of the timestamp, use defines #timestamp as follows in the Custom CSS, Inline attribute. <code>"#timestamp{display:none;}"</code> See Also: "Enabling Asynchronous Updates (page 24-44)"
text.tic	Dial charts only. Defines the numbers on a dial chart.
line.tic	Dial charts only. Defines the graduation mark that displays directly beneath the number on a dial chart.
#dial	Dial charts only. Defines the value that displays on the dial chart.
#dial.alert	Dial charts only. Defines a value (called an alert value) that renders in a dial chart using a different display.
#dialbackground	Dial charts only. Creates the background of a dial chart.
#dialcenter	Dial charts only. Creates the center of the dial on a dial chart.
#dialbackground-border	Dial charts only. Works with #dialbackground to create specific graphic effect.
#low	Dial charts only. Defines the historical low watermark of the data being displayed on a chart.
#high	Dial charts only. Defines the historical high watermark of the data being displayed on a chart.
#XAxisTitle	Defines the title that appears on the x-axis
#YAxisTitle	Defines the title that appears on the y-axis.
.XAxisValue	Defines the value that appears on the x-axis.
.YAxisValue	Defines the value that appears on the y-axis.
.AxisLabel	Similar to the axis value.
.nodatafound	Defines the text element that displays if no information is available.

Table 24-15 (Cont.) Available SVG Chart CSS Classes

Class	Description
.AxisLine	Indicates zero on charts that have negative values.
.GridLine	Creates the horizontal and vertical lines on the chart.
g.dataholder rect	Applies a blanket style to all data that displays in the chart.
.legenditem rect	Applies a blanket style to all rectangular items in the legend.

24.2.6.3 Referencing a Custom SVG Chart Cascading Style Sheet

You can reference a custom cascading style sheet for a chart using the CSS section of the Chart Attributes page. When you reference an external CSS, you can reference it entirely or simply override specific styles.

To reference a custom chart CSS:

1. Upload the CSS to App Builder.
2. Navigate to the page containing the chart.
3. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Chart Attributes**.
4. On the Chart Attributes page, scroll down to the CSS section.
5. From Use Custom CSS, select **Yes**.
6. To reference an external CSS exclusively:
 - a. In Custom CSS, Link, enter a link to a custom CSS. For example:
`#IMAGE_PREFIX#themes/theme_4/svg.css`
 - b. Specify that the CSS should be used exclusively. In Custom CSS, Inline enter the following:
`/**/`
7. To reference a custom CSS and override specific styles:
 - a. In Custom CSS, Link, enter a link to a custom style sheet. For example:
`#IMAGE_PREFIX#themes/theme_4/svg.css`
 - b. In Custom CSS, Inline, enter the custom CSS styles you want to override.

See Also:

- ["About Uploading Cascading Style Sheets \(page 11-81\)"](#)
- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.2.6.4 Specifying Custom CSS Styles Inline for SVG Charts

You can override specific styles within the default CSS, using the Custom CSS, Inline attribute on the Chart Attributes page.

To override specific styles within the default CSS:

1. Create a chart.
2. Navigate to the page.
3. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Chart Attributes**.
4. Scroll down to CSS.
5. From Use Custom CSS, select **Yes**.
6. In Custom CSS, Inline, enter the custom CSS styles you want to override.

See Also:

- ["Understanding SVG Chart Support \(page 24-45\)"](#)
- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.2.6.5 Displaying SVG Charts in Other Languages

To display an SVG chart in another language, you edit the `text` and `tspan` classes to reflect the correct language.

To display an SVG chart in another language:

1. Navigate to the page.
2. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Chart Attributes**.
3. Scroll down to CSS.
4. From Use Custom CSS, select **Yes**.
5. In Custom CSS, Inline, edit the `text` and `tspan` classes to reflect the correct language. The following example demonstrates how to change a chart to Korean:

```
text{font-family:Batang;fill:#000000;}  
tspan{font-family:Batang;fill:#000000;}
```

**See Also:**

["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.2.7 Understanding HTML Chart Support

Previous releases of Oracle Application Express supported the creation of HTML charts. Although existing HTML charts are still supported, Oracle Application Express no longer supports the creation of new HTML charts with the Create wizards. A new HTML chart region type plug-in is available for download from OTN. This plug-in enables users to create HTML chart regions in their applications.

- [Upgrading an HTML Chart to HTML5 \(page 24-51\)](#)
- [Upgrading All HTML Charts to HTML5 \(page 24-52\)](#)

**See Also:**

"Region Type Plug-ins, HTML Chart at:

<http://www.oracle.com/technetwork/developer-tools/apex/application-express/apex-plug-ins-182042.html#region>

24.2.7.1 Upgrading an HTML Chart to HTML5

To upgrade an HTML chart to HTML5:

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. Navigate to the page.
4. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Chart Attributes**.
5. From the Tasks list, click **Upgrade to HTML5 Chart**.
6. Click **Upgrade**.

**See Also:**

["Viewing a Page in Legacy Component View \(page 24-10\)"](#)

24.2.7.2 Upgrading All HTML Charts to HTML5

To upgrade all HTML charts to HTML5:

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. Click **Utilities** and click **Upgrade Application**.
The Upgrade Application Summary report appears.
4. For Upgrade HTML Charts to HTML5 Charts, click the number of candidate objects.
5. Select the objects to upgrade and click **Upgrade**.

24.2.8 About Saving Flash Charts

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.

24.2.9 Using Custom XML with Flash and HTML5 Charts

There are additional chart settings that cannot be controlled using the standard attributes on the Chart attributes page. To further control the look and feel of a chart, you can use custom XML.

To use custom XML:

1. Access the Chart Attributes page:
 - a. Under Region, select the region name.
The Region Definition appears.
 - b. Click **Chart Attributes**.
2. Scroll down to Chart XML.
3. From Use Custom XML, select **Yes**.

If you select to use custom XML, regions with attributes that no longer apply are hidden. To display these regions again, select **No** for Use Custom XML.

4. Edit the XML.
5. Click **Apply Changes**.

 **Tip:**

For more information on supported chart XML formats, see the Online XML Reference at:

<http://6.anychart.com/products/anychart/docs/>

24.3 Creating a Legacy Master Detail Form

 **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](#) (page 9-14)."

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.
 - b. Select a page type - Select **Legacy Page**.
 - c. Click **Next**.
5. Select **Legacy Master Detail** and click **Next**.

The Create Master Detail Wizard appears.

6. 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**.
7. 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.
8. 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**.
9. 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**.
10. 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** if you already have a report page.
 - e. Click **Next**.
11. 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**.
12. 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**.
13. On Navigation Menu, select how you want this page integrated into the Navigation Menu and click **Next**. See field-level Help for more information.
14. On Confirm:
Verify the attributes and confirm your selections.
15. Click **Create**.

 **See Also:**

["Managing Master Detail Forms \(page 9-14\)"](#)

24.4 Managing Legacy Tabular Forms

 **Note:**

A tabular form is a legacy application component. Although still supported, Oracle recommends developers use 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 \(page 24-56\)](#)
- [About Running a Tabular Form \(page 24-57\)](#)
- [How Tabular Forms Work \(page 24-58\)](#)
- [Creating a Legacy Tabular Form \(page 24-59\)](#)
- [Synchronizing Tabular Form Column Order with Region Source \(page 24-61\)](#)

 **See Also:**

- ["Upgrading an Application to Include New Components \(page 6-76\)"](#)
- ["Managing Interactive Grids \(page 8-13\)"](#)
- ["Developing Forms \(page 9-1\)"](#)

24.4.1 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 an 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 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).

24.4.2 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 ▶](#)

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*

24.4.3 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 process.

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.

24.4.4 Creating a Legacy Tabular Form

Use the Create Page Wizard to add a tabular form to an existing application. Once you provide a table name on which to build the tabular form, the wizard reads the table columns from the database data dictionary. You can either specify a primary key defined on the table, or a ROWID pseudo column, which uniquely identifies a row in a table. If you choose to select a primary key, you can choose between three different primary key source types: trigger, PL/SQL function, and sequence.

Tip:

When creating a tabular form, Oracle recommends using a generic column template and avoiding column report templates.

To create a tabular form using the Create Page Wizard:

Note:

Legacy tabular forms are legacy application components. Although Oracle still supports legacy components, Oracle recommends creating interactive grids 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.
 - b. Select **Legacy Page**.
 - c. Click **Next**.
5. Select **Tabular Form** and click **Next**.
The Create Tabular Form Wizard appears.
6. For Page Attributes:
 - a. Implementation - Select the report implementation type.
 - b. Report 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.
 - c. Report Page Name - Enter the title of the page that will contain the report.
 - d. Form Page Mode - Identify the page mode. See field-level Help for more information.

- 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, select how you want this page integrated into the Navigation Menu and click **Next**. See field-level Help for more information.
8. For Data Source:
 - a. Table/View Owner - Select the table or view owner on which to base the tabular form.
 - b. Table/View Name - Select the table or view which contains the columns to be included in the tabular form region.
 - c. Select Column(s) to be shown in Report - 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 **Next**.

Next, define a primary key. Tabular 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.

9. For Form Page, select one of the following for Primary Key :
 - **Managed by Database (ROWID)** - The tabular form uses the ROWID pseudo column to identify rows to update and delete.
 - **Select Primary Key Column(s)** - The tabular form uses the source table's primary key column(s).
10. If you select **Managed by Database (ROWID)**, click **Next** to continue.
11. If you select **Select Primary Key Column(s)**:
 - a. Primary Key Column 1 - Select the first column of the primary key.
 - b. Primary Key Column 2 - If you have a compound primary key, specify the second primary key column in this attribute. This attribute is optional.
 - c. For Source for Primary Key Column1, select the method by which the primary key is populated:
 - **Existing trigger** - Select this option if a before-insert trigger populates the primary key. You can also select this option if you plan on specifying the primary key column source later after completing the form.
 - **Existing sequence** - Select this option to pick the sequence from a list of sequences available in the selected schema.
 - d. Click **Create**.

 **See Also:**

- ["Report Templates \(page 11-72\)"](#)
- ["Managing Interactive Grids \(page 8-13\)"](#)

24.4.5 Synchronizing Tabular Form Column Order with Region Source

To synchronize a classic report's column order with the column order defined in the region source query:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Select a page containing the tabular form.
4. View the page in Component View.
5. To access the Report Attributes page, click the **Report** link next to the Tabular Form region.
6. From Tasks, select **Synchronize Column Order with Region Source**.
7. Click **Synchronize**.

 **See Also:**

["Viewing a Page in Legacy Component View \(page 24-10\)."](#)

24.5 Managing Legacy Calendars

 **Tip:**

Legacy calendars are legacy application components. Although Oracle still supports legacy components, Oracle recommends creating new calendars. In the current release, calendar is based on the FullCalendar jQuery library.

Legacy calendars are template-based. In previous releases Legacy Calendar was referred to as Calendar (Template, Classic calendar, Easy calendar, and SQL calendar).

- [Creating a Legacy Classic Calendar \(page 24-62\)](#)
- [Managing Calendars in Legacy Component View \(page 24-64\)](#)
- [About Supported Calendar Substitution Strings in Legacy Calendars \(page 24-67\)](#)

 **See Also:**

["Creating Calendars \(page 10-1\)."](#)

24.5.1 Creating a Legacy Classic Calendar

 **Tip:**

Legacy calendars are legacy application components. Although Oracle still supports legacy components, Oracle recommends creating new calendars. In the current release, calendar is based on the FullCalendar jQuery library.

 **Tip:**

When creating a Legacy Calendar by running the Create Page Wizard, you must manually create an On Demand page process to enable drag and drop functionality.

To create a Legacy Calendar by running the Create Page Wizard:

 **Tip:**

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

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 **Legacy Page**.
 - c. Click **Next**.
5. Select **Legacy Calendar** and click **Next**.
The Create Calendar Page Wizard appears.
6. 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 - Enter a title for the page to be created.
 - c. Page Mode - Select a page mode. See field-level Help for more information.
 - d. Region Template - Select a region template for the new calendar region. The template controls the appearance of the region.
 - e. Region Name - Enter a name for the region to contain the calendar.
 - f. Display Type - Determines the behavior of the calendar. Options include:

- **Standard** - Submits the page along with the associated request.
- **Partial Page Refresh** - Only refreshes the Calendar region.

 **Tip:**

Once the calendar is created, the display type cannot be changed. To change the display type, you must delete the calendar and then re-create it.

- g. Breadcrumb - Select whether to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
 - h. Click **Next**.
7. For Navigation Menu, 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.
 8. For Source, select if the calendar source is a table or SQL query.

If the Calendar source is **Table**:

- a. Owner - Select the database schema that owns the table from which you are creating a calendar.
- b. Table/View Name - Specify the table to be used in the calendar.
- c. Click **Next**.

If the Calendar source is **SQL Query**:

- a. SQL Query - Enter the SQL SELECT statement to be used by the calendar. The SQL SELECT statement must include at least two columns, of which one must be a DATE column.

For examples, See field-level Help for more information.

- b. Click **Next**.
9. For Calendar Attributes:
 - a. Date Column - Specify which column is used as the date to place an entry on the calendar.
 - b. Display Column - Specify the column to be displayed on the calendar.
 - c. Date Format - Specify how you want to represent dates on your calendar. Options include:
 - **Date Only** - Results display in a monthly and list calendar.
 - **Date and Time** - Results display in a monthly, weekly, daily, and list calendar.
 - d. Custom Date Range - Identify whether to include custom start and end dates. Use this option to enable more than one month to be represented on a monthly calendar.
 - e. Click **Next**.

Next, define Link Details for calendar entries. The Link Target determines where users are redirected to if they select a calendar entry.

10. For Link Details:

- a. Page - Identifies the currently selected page.
 - b. Link Target - Identify whether a calendar entry should have a link target. When a calendar entry is selected, the user is redirected to the link target. The selected Link Target determines what additional attributes display.
See field-level Help for more information.
 - c. Open Link in - Specify the location where the URL or Page should be opened.
 - d. Click **Next**.
11. On Process Option:
- a. Page - Identifies the page number.
 - b. Table / View Name - Identifies the name of the table on which the Form is based on.
 - c. Cancel Button Label - Enter the label for the Cancel button.
 - d. Show Create Button - Indicate whether users should have insert capability using the form.
 - e. Create Button Label - Enter the label for the Create button.
 - f. Show Save Button - Indicate whether users should have update capability using the form.
 - g. Save Button Label - Enter the label for the Save button.
 - h. Show Delete Button - Indicate whether users should have delete capability using the form.
 - i. Delete Button Label - Enter the label for the Delete button.
12. Click **Next**.
13. On Confirm :
- Verify the attributes and confirm your selections.
14. Click **Create**.



See Also:

- ["Creating Calendars \(page 10-1\)"](#)
- ["Editing an Existing Calendar to Include Add and Edit Functionality \(page 24-66\)"](#)
- ["Viewing Field-Level Help \(page 1-27\)."](#)

24.5.2 Managing Calendars in Legacy Component View



Important:

Legacy Component View is disabled by default. To enable Legacy Component View, edit your user preferences.

Once you create a legacy calendar, you can alter the display by editing attributes. 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 Component View](#) (page 24-65)
- [Editing an Existing Calendar to Include Add and Edit Functionality](#) (page 24-66)
- [Enabling the Dragging and Dropping of Data In an Existing Legacy Calendar](#) (page 24-67)

**See Also:**

["Editing User Preferences](#) (page 1-23)"

24.5.2.1 Editing Calendar Attributes in Component View

Once you create a calendar, you can alter the display by editing attributes. 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.

**Tip:**

To disable a view of a calendar, you must delete the Monthly, Weekly, or Daily buttons on the calendar page.

To access Calendar attributes:

1. Navigate to the appropriate page. See .
2. Access the Region Definition page. Under Region, select the region name.
The Region Definition appears. Use these attributes to alter the container that holds the calendar, including the region title.
3. Click **Calendar Attributes**.
The Calendar Attributes page appears.
4. Edit the appropriate attributes.
To learn more about an attribute, see field-level Help.
5. Click **Apply Changes**.

 **See Also:**

- ["Viewing a Page in Legacy Component View \(page 24-10\)"](#)
- ["Viewing Field-Level Help \(page 1-27\)."](#)

24.5.2.2 Editing an Existing Calendar to Include Add and Edit Functionality

When defining calendar create and edit links for Desktop applications, Oracle recommends defining two separate pages. For the page used for creating a new calendar entry the Page Access Protection should be defined as **Unrestricted**. However, for the page used for editing existing calendar entries the Page Access Protection should be set to **Arguments Must Have Checksum**.

To edit an existing calendar to include add and edit functionality:

1. Navigate to the appropriate Calendar Attributes page.
2. Scroll down to Create Link or View / Edit Link. Enter the URL.

For example, a Create Link for a duration based event navigating to Page 3 would be:

```
f?p=&APP_ID.:3:&SESSION.::&DEBUG.:RP,  
3:P3_START_DATE,P3_END_DATE:#APEX$NEW_START_DATE#, #APEX$NEW_END_DATE#:
```

For example, a View / Edit Link using ROWID for the primary key column, navigating to Page 4 would be:

```
f?p=&APP_ID.:4:&SESSION.::&DEBUG.:RP,4:P4_ROWID:&ROWID.:
```

 **Tip:**

The selected date column must have a format mask of `RRRRMMDDHH24MISS` since the calendar passes the date in that format.

If you do not want to change the format mask of date item, you must create a hidden item (this hidden item should be selected as Date Item which accepts the date value) in the form page with the above format mask.

3. Scroll up to Calendar Display.

From Primary Key Column, select the value to be passed when editing calendar data. This value is substituted for the `#PRIMARY_KEY_VALUE#` specified in the previous step.

4. Click **Apply Changes**.

 **See Also:**

- ["Editing Calendar Attributes in Component View \(page 24-65\)"](#)

24.5.2.3 Enabling the Dragging and Dropping of Data In an Existing Legacy Calendar

When you create a Legacy calendar on a new page, the Drag and Drop option does not appear. To enable this functionality, you must create the On Demand process, create a Legacy calendar page, and then edit the page to enable drag and drop. As a best practice, Oracle recommends creating this process before creating the calendar.

To enable support for dragging and dropping of calendar data:

1. Create an On Demand application process.

This process should execute when the calendar data is dragged and dropped. It should have code similar to the following example:

```
declare
  l_date_value      varchar2(32767) := apex_application.g_x01;
  l_primary_key_value varchar2(32767) := apex_application.g_x02;
begin
  update EMP set HIREDATE = to_date(l_date_value, 'YYYYMMDD HH24MISS')
  where ROWID = l_primary_key_value;
end;
```

2. Navigate to the Calendar Attributes.
3. Scroll down to Drag and Drop.
4. From Process, select the custom page process you created in step 1.
5. Click **Apply Changes**.

 **See Also:**

["Understanding Application Processes \(page 15-32\)"](#)

24.5.3 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 \(page 11-1\)"](#)
- ["Creating a Legacy Classic Calendar \(page 24-62\)"](#)

24.6 Viewing Legacy Components in Gallery

The following legacy components do not display in gallery by default:

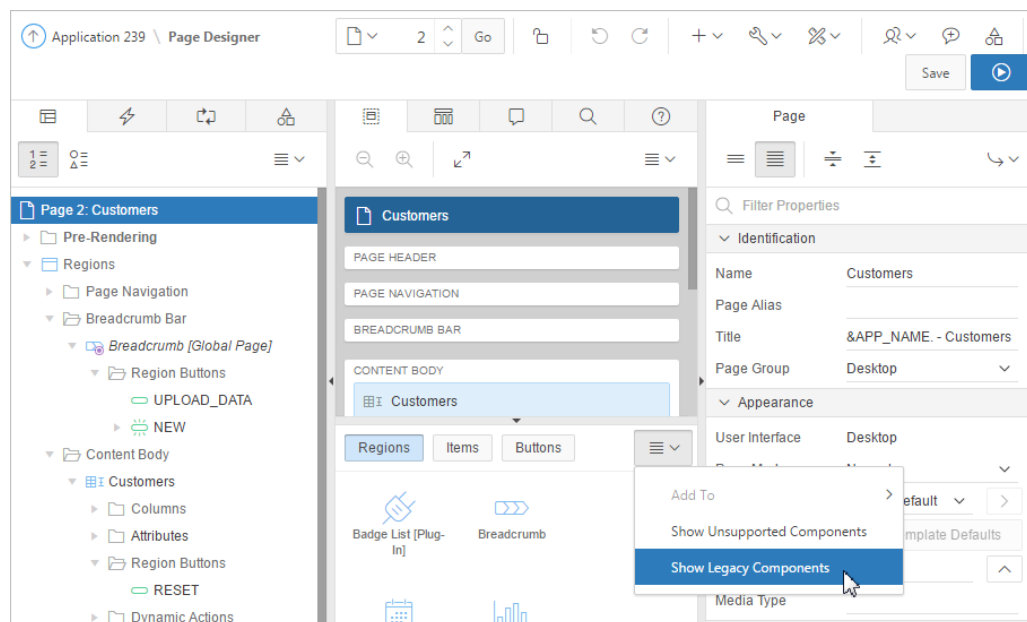
- Legacy Tabular Form
- Legacy Master Detail Form
- Legacy Classic Calendars
- Legacy AnyChart Charts

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

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. The following table describes key attributes that control item functionality. To learn more about specific item attributes, see field-level Help.

- [Available Item Types](#) (page A-1)

See Also:

- [Viewing Field-Level Help](#) (page 1-27)
- [Understanding Page-Level Items](#) (page 13-1)
- [Editing Page-Level Items in Page Designer](#) (page 13-14)
- [Editing Page-Level Items in Component View](#) (page 13-28)

A.1 Available Item Types

[Table A-1](#) (page A-2) 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 (<code>ename</code>), but returns employee numbers (<code>empno</code>):</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 (page 13-13)"</p> <p>Component View item attributes:</p> <ul style="list-style-type: none"> • Settings, 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 of 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 (page 13-25)"</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>Component View item attributes:</p> <ul style="list-style-type: none"> • Settings, 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 (page 16-10)"</p>

Table A-1 (Cont.) Available Item Types

Item Type	Description
Date Picker	<p data-bbox="521 331 1458 422">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 data-bbox="521 428 1458 485">Time is only displayed in the calendar popup if the Format Mask for this item includes time components.</p> <p data-bbox="521 491 846 518">Page Designer item attributes:</p> <ul data-bbox="521 525 1458 957" style="list-style-type: none"> <li data-bbox="521 525 1458 581">• Settings, Highlighted Date - Enter a default date value to be highlighted when the calendar pop-up displays. See Help for more details. <li data-bbox="521 588 1458 615">• Settings, Minimum Date - Enter the minimum date that can be input or selected. <li data-bbox="521 621 1458 648">• Settings, Maximum Date - Enter the maximum date that can be input or selected. <li data-bbox="521 655 1458 682">• Settings, Show - Select when the date picker pop-up calendar displays. <li data-bbox="521 688 1458 772">• 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. <li data-bbox="521 779 1458 835">• Settings, Navigation List for- Select whether select lists are provided within the calendar pop-up for the month and/or year. <li data-bbox="521 842 1458 869">• Appearance, Format Mask - Enter a date format mask to apply to this component. <li data-bbox="521 875 1458 957">• 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 data-bbox="521 963 870 991">Component View item attributes:</p> <ul data-bbox="521 997 1458 1514" style="list-style-type: none"> <li data-bbox="521 997 1458 1081">• Settings, 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. <li data-bbox="521 1087 1458 1171">• Settings, Format Mask - Identify the date format mask used by the item type. You can use substitution syntax to reference a dynamic format mask from an application or page item. <li data-bbox="521 1178 1458 1262">• Settings, Highlighted Date - Enter a default date value to be highlighted when the calendar pop-up is invoked. If not specified, the calendar highlights the current date and time. See Help for more details. <li data-bbox="521 1268 1458 1295">• Settings, Minimum Date - Enter the minimum date that can be input or selected. <li data-bbox="521 1302 1458 1329">• Settings, Maximum Date - Enter the maximum date that can be input or selected. <li data-bbox="521 1335 1458 1362">• Settings, Show - Select when the date picker pop-up calendar displays. <li data-bbox="521 1369 1458 1453">• 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. <li data-bbox="521 1459 1458 1514">• Settings, Navigation List - Select whether select lists are provided within the calendar pop-up for the month and/or year.
Display Image	<p data-bbox="521 1520 1458 1556">Displays an image stored in a database BLOB columns, or based on an image URL.</p> <p data-bbox="521 1562 1081 1589">Page Designer and Component View item attributes:</p> <ul data-bbox="521 1596 1458 1923" style="list-style-type: none"> <li data-bbox="521 1596 1458 1652">• Settings, Based On - Determines where the item is retrieved from. To learn more about an attribute, see field-level Help. <li data-bbox="521 1659 1458 1743">• 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. <li data-bbox="521 1749 1458 1806">• Settings, Filename Column - Enter the uppercase page or application item name that holds the filename column value associated with the BLOB column. <li data-bbox="521 1812 1458 1923">• 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.

Table A-1 (Cont.) Available Item Types

Item Type	Description
Display Only	<p>Displays a non-enterable text item.</p> <p>Page Designer and Component View 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>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>See Also: "About Securing File Uploads (page 19-39)"</p> <p>Page Designer attributes:</p> <ul style="list-style-type: none"> • Settings, Storage Type - Identify where the uploaded file is stored. Options include: <ul style="list-style-type: none"> – 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. – 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>If Storage Type equals Table APEX_APPLICATION_TEMP_FILES:</p> <ul style="list-style-type: none"> – Purge File at - Define when Application Express should remove the temporary file. – 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.. – 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>If Storage Type equals BLOB column specified in Item Source:</p> <ul style="list-style-type: none"> – 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. – Filename Column - Enter the page or application item used to store the filename of the uploaded file. . – 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. – 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.. – Display Download Link - Specify whether the download link displays when there is a non-empty BLOB associated with this item. – 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. – Content Disposition - Select how the browser handles the content when downloading. • 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. <p>Component View attributes:</p> <ul style="list-style-type: none"> • Settings, 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. • Settings, Storage Type - Identify where the uploaded file is stored.

Table A-1 (Cont.) Available Item Types

Item Type	Description
	<ul style="list-style-type: none"> • Settings, MIME Type Column - Specify the name of the column where the <code>MIME</code> type of the uploaded file should be stored. • Settings, Filename Column - Specify the name of the column where the file name of the uploaded file should be stored. • Settings, Character Set Column - Specify the name of the column where the character set of the uploaded file should be stored. • Settings, BLOB Last Updated Column - Specify the name of the column to update with the current time stamp when a file is uploaded. • Settings, Display Download Link - Select Yes to display the download link next to the Browse button. • Settings, Download Link Text - Text used for the download link. • Settings, Content Disposition - Determines if the downloaded content is displayed inline within the Browser window or if a Open dialog displays.
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 and Component View 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 (page 13-13)"</p> <p>Component View attributes:</p> <ul style="list-style-type: none"> • Settings, 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. • Settings, Fetch - Select how data is displayed when the Popup LOV is opened. <p>See Also: "Creating a Cascading List of Values (page 13-25)"</p>

Table A-1 (Cont.) Available Item Types

Item Type	Description
Number Field	<p data-bbox="524 331 1458 422">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 data-bbox="524 428 797 455">Page Designer attributes:</p> <ul data-bbox="524 462 1458 758" style="list-style-type: none"><li data-bbox="524 462 1458 520">• Settings, Minimum Value - Enter the minimum value permitted. The value can be a static value or a dynamic value using substitution syntax.<li data-bbox="524 527 1458 585">• Settings, Maximum Value - Enter the maximum value permitted. The value can be a static value or a dynamic value using substitution syntax.<li data-bbox="524 592 1458 674">• 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.<li data-bbox="524 680 1458 758">• 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 data-bbox="524 764 818 791">Component View attributes:</p> <ul data-bbox="524 798 1458 1190" style="list-style-type: none"><li data-bbox="524 798 1458 888">• Settings, 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.<li data-bbox="524 894 1458 976">• Settings, Format Mask - Identify the number format mask used by the item type. You can use substitution syntax to reference a dynamic format mask from an application or page item.<li data-bbox="524 982 1458 1041">• Settings, Minimum Value - Enter the minimum value permitted. The value can be a static value or a dynamic value using substitution syntax.<li data-bbox="524 1047 1458 1106">• Settings, Maximum Value - Enter the maximum value permitted. The value can be a static value or a dynamic value using substitution syntax.<li data-bbox="524 1113 1458 1190">• 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.

Table A-1 (Cont.) Available Item Types

Item Type	Description
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>Component View attributes:</p> <ul style="list-style-type: none"> • Settings, 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, 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> <p>See Also: "About Session State and Security (page 19-26)".</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>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>SELECT ename ' ' job display_value, empno FROM emp</pre> <p>Page Designer item attributes:</p> <ul 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>Component View item attributes:</p> <ul style="list-style-type: none">• Settings, 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, Input Field - Select whether the item is enterable.• Settings, Fetch - Select how data is initially fetched for the list of values. <p>See Also: "Creating a Cascading List of Values (page 13-13)" and "Creating a Cascading List of Values (page 13-25)"</p>

Table A-1 (Cont.) Available Item Types

Item Type	Description
Radio Group	<p data-bbox="444 338 1377 390">Displays multiple values as radio group options, enabling the end user to select a single value.</p> <p data-bbox="444 405 1377 457">The following example displays employee names (<code>ename</code>), but returns employee numbers (<code>empno</code>):</p> <pre data-bbox="444 485 781 510">SELECT ename, empno FROM emp</pre> <p data-bbox="444 537 768 562">Page Designer item attributes:</p> <ul data-bbox="444 577 1377 1077" style="list-style-type: none"> <li data-bbox="444 577 1377 657">• 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. <li data-bbox="444 667 1377 720">• Settings, Page Action on Selection - Select what action is taken when a radio group value is selected. See Help for more details. <li data-bbox="444 730 1377 926">• 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 (page 15-19)." See Also: "Creating a Cascading List of Values (page 13-13)" and "Creating a Cascading List of Values (page 13-25)" <li data-bbox="444 999 1377 1077">• 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 data-bbox="444 1087 792 1113">Component View item attributes:</p> <ul data-bbox="444 1127 1377 1719" style="list-style-type: none"> <li data-bbox="444 1127 1377 1207">• Settings, 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. <li data-bbox="444 1218 1377 1297">• 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. <li data-bbox="444 1308 1377 1388">• Settings, Page Action on Selection - Select what action is taken when a radio group value is selected. Option includes: None (Default); Submit Page; or Redirect and Set Value. Choose Submit Page to have the page submitted when the radio button is selected. <li data-bbox="444 1461 1377 1656">• 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 (page 15-19)." See Also: "Creating a Cascading List of Values (page 13-13)" and "Creating a Cascading List of Values (page 13-25)"

Table A-1 (Cont.) Available Item Types

Item Type	Description
Rich Text Editor	<p data-bbox="521 331 1458 447">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 data-bbox="521 457 846 485">Page Designer item attributes:</p> <ul data-bbox="521 495 1458 825" style="list-style-type: none"> <li data-bbox="521 495 1458 548">• Settings, Editor - Select the editor used to display the text formatting options available. <li data-bbox="521 558 1458 611">• Settings, Toolbar - Select the number of text formatting options available within the editor. <li data-bbox="521 621 1458 674">• Settings, Toolbar Expanded - Specify whether the editor is initially expanded within the text area. <li data-bbox="521 684 1458 716">• Settings, Skin - Select the editor skin used to style the editor. <li data-bbox="521 726 1458 758">• Settings, Toolbar Position - Specify where in the text area the editor displays. <li data-bbox="521 768 1458 825">• 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 data-bbox="521 835 870 863">Component View item attributes:</p> <ul data-bbox="521 873 1458 1203" style="list-style-type: none"> <li data-bbox="521 873 1458 957">• Settings, 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. <li data-bbox="521 968 1458 1020">• Settings, Editor - Select the editor used to display the text formatting options available. <li data-bbox="521 1031 1458 1083">• Settings, Toolbar - Select the number of text formatting options available within the editor. <li data-bbox="521 1094 1458 1146">• Settings, Toolbar Expanded - Determines if the toolbar must start expanded when the Rich Text Editor is loaded. <li data-bbox="521 1157 1458 1188">• Settings, Skin - Select the editor skin used to style the editor. <li data-bbox="521 1199 1458 1203">• Settings, Toolbar Position - Specify where in the text area the editor displays.

Table A-1 (Cont.) Available Item Types

Item Type	Description
Select List	<p data-bbox="444 338 1377 394">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 data-bbox="444 405 1377 489">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="444 510 708 646"> select [displayValue], [returnValue] from ... where ... order by ... </pre> <p data-bbox="444 678 764 703">Page Designer item attributes:</p> <ul data-bbox="444 716 1377 1066" style="list-style-type: none"> <li data-bbox="444 716 1377 772">• Settings, Page Action on Selection - Select what action is taken when a list value is selected. See help for options. <li data-bbox="444 783 1377 982">• 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 (page 15-19)." <li data-bbox="444 993 1377 1066">• 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 data-bbox="444 1077 792 1102">Component View item attributes:</p> <ul data-bbox="444 1115 1377 1455" style="list-style-type: none"> <li data-bbox="444 1115 1377 1192">• Settings, 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. <li data-bbox="444 1203 1377 1255">• Settings, Page Action on Selection - Select what action is taken when a list value is selected. <li data-bbox="444 1266 1377 1455">• 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 (page 15-19)."

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 (page 16-24)," "Working with Multiple Select List Item (page 13-4)," and "Creating a Cascading List of Values (page 13-13)."</p> <p>Component View item attributes:</p> <ul style="list-style-type: none"> • Settings, 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, Show Controls - Select what data movement controls displays in the middle of the shuttle. Options include: None, Moving Only, Ordering Only, or All. <p>Note: To create this item type, you must define a list of values. See "Creating Lists of Values at the Application Level (page 16-24)," "Working with Multiple Select List Item (page 13-4)," and "Creating a Cascading List of Values (page 13-25)."</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>Component View item attributes:</p> <ul style="list-style-type: none"> • Settings, 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, 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 <p>See Also: "Configuring Switch Defaults (page 16-11)"</p>

Table A-1 (Cont.) Available Item Types

Item Type	Description
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. <p>Component View item attributes:</p> <ul style="list-style-type: none">• Settings, 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, 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.

Table A-1 (Cont.) Available Item Types

Item Type	Description
Text Field	<p data-bbox="521 338 862 363">Displays the item as a text field.</p> <p data-bbox="521 373 846 399">Page Designer item attributes:</p> <ul data-bbox="521 409 1456 735" style="list-style-type: none"> <li data-bbox="521 409 1456 556">• 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. 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. <li data-bbox="521 651 1456 735">• 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 data-bbox="521 745 870 770">Component View item attributes:</p> <ul data-bbox="521 781 1456 1304" style="list-style-type: none"> <li data-bbox="521 781 1456 865">• Settings, 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. <li data-bbox="521 875 1456 1022">• 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. <li data-bbox="521 1022 1456 1075">• Settings, Trim Spaces - Select how the item value is trimmed after submitting the page. This setting trims spaces, tabs, and new lines from the text entered. <li data-bbox="521 1085 1456 1138">• Settings, Submit when Enter Pressed - Specify whether pressing the Enter key while in this field automatically submits the page. <li data-bbox="521 1148 1456 1304">• Settings, Disabled - Specify whether this item is disabled, which prevents end users from changing the value. A disabled text item still displays with the same HTML formatting, unlike an item type of Display Only, which removes the HTML formatting. Disabled text items are part of page source, which enables their session state to be evaluated. Conversely, display only items are not stored in session state.

Table A-1 (Cont.) Available Item Types

Item Type	Description
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="444 447 708 583"> select [displayValue], [returnValue] from ... where ... order by ... </pre> <p>Page Designer item attributes:</p> <ul data-bbox="444 653 1383 1388" 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. <p>Component View item attributes:</p> <ul data-bbox="444 1434 1383 1929" style="list-style-type: none"> • Settings, 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, 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 - Specify whether the text field is filled automatically with the first autocomplete value. • Settings, Automatic Fill - Specify whether the text field is filled automatically with the first autocomplete value. See Help for more details.

Table A-1 (Cont.) Available Item Types

Item Type	Description
Text Area	<ul style="list-style-type: none"> • 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. <p>Displays a multiple-row text area.</p> <p>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>Page Designer item attributes:</p> <ul style="list-style-type: none"> • Settings, Resizable - Specify whether the text area resizing using the mouse is enabled. • 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. • Settings, Character Counter - Select Yes to include a counter that displays the number of characters entered in the field. • Validation, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a <code>NOT NULL</code> validation when the page is submitted. <p>Component View item attributes:</p> <ul style="list-style-type: none"> • Settings, Value Required- If select Yes and the page item is visible, Oracle Application Express automatically performs a <code>NOT NULL</code> validation when the page is submitted. • Settings, Resizable - Specify whether the text area resizing using the mouse is enabled. • 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. • Settings, Character Counter - Select Yes to include a counter that displays the number of characters entered in the field. • Settings, Trim Spaces - Select how the item value is trimmed after submitting the page. This setting trims spaces, tabs, and new lines from the text entered.

B

Oracle Application Express Limits

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. [Table B-1](#) (page B-1) describes limits associated with Oracle Application Express components.

- [Component Limits](#) (page B-1)

B.1 Component Limits

[Table B-1](#) (page B-1) 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
Interactive reports	999 rows per column heading filter (if no custom LOV is specified in the column attributes). See Also: " Developing Reports (page 8-1)"
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. See Also: " Developing Forms (page 9-1)"
Tabular Forms	One wizard-generated tabular form per page (using built-in DML). 50 editable tabular form columns (<code>apex_application.g_f01-apex_application.g_f50</code>), generated with <code>apex_item</code> or the built-in tabular form display types. See Also: " Managing Legacy Tabular Forms (page 24-56)"
Item names	Item names should not exceed 30 characters. Items longer than 30 characters cannot be referenced using bind variable syntax. See Also: " Referencing Session State Using Bind Variable Syntax (page 2-24)" and " About Item Naming Conventions (page 13-2)"
Validations	Text entered for validations may not exceed 3,950 characters. See Also: " Understanding Validations (page 15-8)"
Data loading	When loading data, the maximum number of columns is 45. See Also: " Creating Applications with Data Loading Capability (page 17-1)"

C

Available Conditions

App Builder includes a number of condition that enable developers to 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.

- [Conditions](#) (page C-1)
- [Conditions Available in App Builder](#) (page C-1)

C.1 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" \(page 2-10\)](#) and ["Conditions Available in App Builder" \(page C-1\)](#)

C.2 Conditions Available in App Builder

[Table C-1](#) (page C-1) 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.

Table C-1 (Cont.) Available Conditions

Condition	Description
Request = Value	The page request is equal to the text you enter into the Value attribute.
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.

Table C-1 (Cont.) Available Conditions

Condition	Description
Page/Region is NOT Read Only	The page and parent region, if appropriate, are displayed normally.
User is authenticated (not public)	The end user is logged into the application. See Also: " Establishing User Identity Through Authentication (page 19-45)" 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

This section describes grant privileges in Oracle Application Express.

- [About Granting Privileges to Database Users](#) (page D-1)
- [Understanding Privileges Granted to PUBLIC](#) (page D-1)

D.1 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]
```

D.2 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_050100).

- [Packages](#) (page D-2)

- [Procedures](#) (page D-4)
- [Functions](#) (page D-4)
- [Tables](#) (page D-4)
- [Views](#) (page D-5)
- [Sequences](#) (page D-9)
- [Types](#) (page D-9)

D.2.1 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_050100):

```
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_050100):

WWV_CALCULATOR
WWV_FLOW_FLASH_CHART

WWV_FLOW_GENERATE_DDL
 WWV_FLOW_IMAGE_GENERATOR
 WWV_FLOW_TREE_GLOBAL_VARS

D.2.2 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_050100):

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

D.2.3 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_050100):

NV
 V

Execute privilege is granted to PUBLIC for the following function owned by the Oracle Application Express product schema, (for example, apex_050100):

WWV_POPUP_FILTER

D.2.4 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_050100):

WWV_FLOW_DUAL100

WWV_FLOW_TEMP_TABLE

WWV_FLOW_LOV_TEMP

D.2.5 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_050100):

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_SHORTCUTS
APEX_APPLICATION_SUBSTITUTIONS
APEX_APPLICATION_SUPP_OBJECTS
APEX_APPLICATION_SUPP_OBJ_BOPT
APEX_APPLICATION_SUPP_OBJ_CHCK
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_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_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

D.2.6 Sequences

The following sequences have public synonyms and select privilege granted to public:

WWV_FLOW_SESSION_SEQ
WWV_SEQ

D.2.7 Types

The following types have execute granted to public but no synonyms:

WWV_FLOW_TREE_ENTRY

WWV_FLOW_TREE_SUBS

Index

A

- Accept Page, [2-10](#)
- access control list
 - about, [19-39](#)
 - adding users, [19-43](#)
 - creating, [19-40](#)
 - removing users, [19-44](#)
 - selecting an application mode, [19-42](#)
 - selecting user privileges, [19-43](#)
- access control, authorization schemes, [19-44](#)
- Account menu, [1-10](#)
- account profile
 - changing your password, [1-22](#)
 - editing email address, [1-22](#)
 - editing name, [1-22](#)
 - uploading a photo, [1-22](#)
- activity log, [21-8](#)
- activity reports, creating custom, [21-8](#)
- Administration menu, [1-8](#)
- Administration Menu
 - Administration, [1-8](#)
 - Change My Password, [1-8](#)
 - Dashboards, [1-8](#)
 - Manage Service, [1-8](#)
 - Manage Users and Groups, [1-8](#)
 - Monitory Activity, [1-8](#)
- Allow Feedback attribute, [6-19](#)
- Allows URLs Created After attribute, [6-29](#)
- AnyCo IT Department
 - about, [3-19](#)
 - installing, [3-19](#)
 - navigating, [3-20](#)
 - running, [3-19](#)
- APEX_ACTIVITY_LOG view, [21-8](#)
- APEX_MAIL, [18-3](#)
- App Builder
 - accessing, [1-12](#), [1-16](#)
 - home page, [1-13](#)
 - page, [2-2](#)
- App Builder home page, [1-12](#), [1-13](#), [1-16](#)
 - Create button, [1-15](#)
 - Migration link, [1-16](#)
 - navigation bar, [1-13](#)
 - Recent list, [1-16](#)
- App Builder home page (*continued*)
 - Reset button, [1-15](#)
- App Builder utilities
 - Advisor, [6-74](#)
 - Application Dashboard, [6-74](#)
 - Application Express Views, [6-74](#)
 - Attribute Dictionary, [6-74](#)
 - Change History, [6-74](#)
 - Database Object Dependencies, [6-74](#)
 - Debug Messages, [6-74](#)
 - Export Repository, [6-74](#)
 - Recently Updated Pages, [6-74](#)
 - Upgrade Application, [6-74](#)
- APP_AJAX_X01, [2-31](#)
- APP_AJAX_X10, [2-31](#)
- application
 - adding content from Web sites, [12-30](#)
 - adding Data Load Wizard, [17-1](#)
 - adding data loading, [17-1](#)
 - attaching an authorization scheme, [19-64](#)
 - changing authentication scheme, [19-58](#)
 - controlling navigation using branches, [14-25](#)
 - copying, [1-21](#)
 - creating, [2-1](#)
 - debugging, [21-1](#), [22-1](#)
 - defining primary language, [20-5](#)
 - deleting, [1-21](#)
 - deploying, [23-4](#), [23-6](#), [23-19](#)
 - exporting, [23-6](#), [23-19](#)
 - grouping, [6-71](#)
 - hiding features, [23-42](#)
 - importing export file, [23-30](#)
 - importing from SQL*Plus, [23-38](#)
 - language preference, [20-2](#)
 - resource use, [22-3](#)
 - translatable components, [20-2](#)
 - translating, [20-1](#)
 - translating multibyte languages, [20-7](#)
 - translation rendering, [20-2](#)
 - upgrading, [6-76](#), [6-77](#)
- Application Alias attribute, [6-18](#)
- application attributes
 - Build Options, [6-23](#)
 - Database Session, [6-31](#)
 - editing, [6-16](#)

- application attributes (*continued*)
 - Error Handling, [6-21](#)
 - exporting, [23-21](#)
 - Substitutions, [6-23](#)
- application computations
 - about, [15-37](#)
 - creating, [15-38](#)
 - executing On New Instance, [15-33](#), [15-38](#)
 - history report, [15-39](#)
- Application Dashboard, [6-75](#)
- Application Dashboard report, [21-4](#)
- Application Date Format attribute, [6-34](#)
- Application Date Time Format attribute, [6-34](#)
- application definition, editing, [6-16](#)
- Application Express Accounts authentication
 - setting up, [19-49](#)
- Application Express Views, [6-78](#)
- application groups, [6-72](#)
- Application Groups attribute, [6-18](#)
- application Help, creating, [10-33](#)
- Application home page
 - adding database application page, [6-48](#)
 - Export/Import icon, [1-19](#)
 - Run Application button, [1-18](#)
 - Shared Components icon, [1-18](#)
 - Supporting Objects icon, [1-18](#)
 - Utilities icon, [1-18](#)
- Application home Task list
 - Copy this Application, [1-21](#)
 - Delete this Application, [1-21](#)
- application item
 - about, [13-2](#), [16-20](#)
 - creating, [16-21](#)
 - creating a global variable, [16-23](#)
 - editing attributes, [16-23](#)
 - history, [16-22](#)
- application language, [6-32](#)
- Application Language Derived From attribute, [6-33](#), [20-5](#)
- application layout
 - print preview mode, [12-2](#)
 - shortcuts, [16-42](#)
- Application Primary Language attribute, [6-33](#), [20-5](#)
- application processes
 - creating, [15-34](#)
 - creating error message, [15-36](#)
 - editing attributes, [15-35](#)
 - example, [15-34](#)
 - On Demand, [15-32](#)
 - reports, [15-36](#)
 - understanding, [15-32](#)
- Application Timestamp Format, [6-34](#)
- Application Timestamp Time Zone Format, [6-34](#)
- application utilities
 - accessing, [6-74](#)
- architecture, [1-2](#)
- asynchronous updates
 - charts, [24-39](#), [24-44](#)
 - maps, [10-20](#), [10-23](#)
- Attribute Dictionary, [15-40](#)
 - about, [15-40](#)
 - accessing, [15-40](#)
- authentication, [2-12](#), [19-45](#)
 - Authentication Schemes Repository, [19-57](#)
 - changing, [19-58](#)
 - creating an authentication scheme, [19-47](#)
 - managing existing schemes, [19-57](#)
 - preconfigured authentication schemes, [19-48](#)
 - reports, [19-59](#)
 - security attributes, [6-24](#)
 - understanding, [19-45](#), [19-46](#)
 - viewing current scheme, [19-58](#)
- authentication report
 - Authentication Scheme History, [19-59](#)
 - Authentication Scheme Subscription, [19-59](#)
- Authentication Scheme attribute, [6-25](#)
- Authentication Scheme History report, [19-59](#)
- Authentication Scheme Subscription report, [19-59](#)
- authentication schemes
 - Application Express Accounts, [19-48](#)
 - cookie user accounts, [19-48](#)
 - Custom authentication, [19-49](#)
 - DAD Credentials, [19-54](#)
 - Database Account Credentials, [19-51](#)
 - HTTP Header Variable, [19-52](#)
 - LDAP Directory Verification, [19-53](#)
 - Open Door Credentials, [19-55](#)
 - Single Sign-On verification, [19-56](#)
- authorization, [2-12](#)
- Authorization attributes, [6-25](#)
- authorization scheme
 - attaching to application, [19-64](#)
 - attaching to component, [19-64](#)
 - attaching to control, [19-64](#)
 - attaching to page, [19-64](#)
 - By Component report, [19-65](#)
 - History report, [19-65](#)
 - of an application, [6-25](#)
 - reports, [19-65](#)
 - specifying, [6-25](#)
 - Subscription report, [19-65](#)
 - Utilization report, [19-65](#)
- Authorization Scheme attribute, [6-25](#)
- Authorization Scheme History report, [19-65](#)
- Authorization Scheme Subscription report, [19-65](#)
- Authorization Scheme Utilization report, [19-65](#)
- authorization schemes, [19-60](#)

- authorization schemes (*continued*)
 - attaching to application, [19-64](#)
 - attaching to component, [19-64](#)
 - attaching to control, [19-64](#)
 - attaching to page, [19-64](#)
 - creating, [19-60](#)
 - for controlling access, [19-44](#)
 - Automatic CSV Encoding attribute, [6-35](#)
 - Automatic Time Zone, [6-35](#)
- ## B
-
- bind variables
 - syntax, [2-24](#)
 - using in regions, [2-25](#)
 - using in regions based on PL/SQL, [2-25](#)
 - blank page
 - creating with Create Application Wizard, [6-3](#)
 - BLOB support in forms
 - about, [9-24](#)
 - displaying the BLOB, [9-25](#)
 - downloading files, [9-24](#)
 - removing the image reference, [9-25](#)
 - uploading files, [9-24](#)
 - BLOB support in reports
 - about, [8-131](#)
 - displaying the BLOB, [8-132](#)
 - providing download link, [8-131](#)
 - BLOBs
 - APEX_UTIL.GET_BLOB_FILE_SRC, [8-133](#)
 - download support, [9-24](#)
 - populating, [9-24](#)
 - support, [8-130](#)
 - support in forms, [9-24](#)
 - working with procedurally, [8-133](#)
 - Bookmark Hash Function attribute, [6-29](#)
 - branch
 - about, [15-32](#)
 - creating, [14-25](#), [14-27](#)
 - page submission, [7-30](#)
 - pre-rendering, [7-30](#)
 - When Button Pressed attribute, [2-44](#)
 - BRANCH_TO_PAGE_ACCEPT, [2-45](#)
 - branching
 - making conditional, [14-28](#)
 - on load, before header, [14-27](#)
 - on submit, after processing, [14-27](#)
 - on submit, before computation, [14-27](#)
 - on submit, before processing, [14-27](#)
 - on submit, before validation, [14-27](#)
 - using buttons, [13-61](#)
 - breadcrumb reports, [14-18](#)
 - Breadcrumb Exceptions, [14-18](#)
 - Breadcrumb Hierarchy, [14-18](#)
 - Breadcrumb History, [14-18](#)
 - breadcrumb reports (*continued*)
 - Breadcrumb Utilization, [14-18](#)
 - Grid Edit Breadcrumb, [14-18](#)
 - breadcrumb template, [11-47](#)
 - Definition, [11-48](#)
 - Name, [11-48](#)
 - Substitution Strings, [11-48](#)
 - breadcrumb template substitution strings
 - #LINK#, [11-49](#)
 - #LONG_NAME#, [11-49](#)
 - #NAME#, [11-49](#)
 - NAME_ESC_SC#, [11-49](#)
 - breadcrumbs
 - about, [14-9](#)
 - about creating manually, [14-10](#)
 - adding a breadcrumb region, [14-14](#), [14-15](#)
 - adding entries, [14-12](#)
 - adding to a page, [14-14](#), [14-15](#)
 - creating, [14-8](#)
 - creating as shared components, [14-11](#)
 - creating dynamic, [14-16](#)
 - creating from Component View, [14-12](#)
 - creating with Create Region Wizard, [14-11](#)
 - deleting unused, [14-17](#)
 - editing, [14-16](#)
 - Exceptions report, [14-18](#)
 - Grid Edit report, [14-18](#)
 - Hierarchy report, [14-18](#)
 - History report, [14-18](#)
 - navigations, [11-47](#)
 - reparenting entries, [14-17](#)
 - Utilization report, [14-18](#)
 - Browse button, [A-5](#)
 - bug reports
 - Bugs Dashboard, [4-19](#)
 - By Day, [4-20](#)
 - By Day page, [4-20](#)
 - By Developer, [4-20](#)
 - By Developer page, [4-20](#)
 - Calendar page, [4-20](#)
 - bugs
 - attaching files, [4-18](#)
 - Bugs page, [4-17](#)
 - creating, [4-17](#)
 - editing, [4-17](#)
 - managing, [4-16](#)
 - viewing, [4-17](#)
 - viewing reports, [4-19](#)
 - build options, [23-40](#)
 - changing status, [23-41](#)
 - creating, [23-41](#)
 - excluding, [23-41](#)
 - exporting, [23-42](#)
 - exporting status, [23-42](#)
 - including, [23-41](#)

- build options (*continued*)
 - reports, [23-43](#)
 - Build Options attribute, [6-23](#)
 - Build Status attribute, [6-20](#)
 - built-in substitution strings, [2-28](#)
 - Bulk Edit Item Help report, [10-38](#)
 - accessing, [10-38](#)
 - button
 - branching, [13-61](#)
 - creating, [13-51](#), [13-52](#), [13-55](#)
 - creating multiple at once, [13-57](#)
 - delete multiple, [13-59](#)
 - displaying conditionally, [13-53](#), [13-59](#)
 - editing, [13-53](#), [13-54](#), [13-58](#)
 - editing multiple, [13-59](#)
 - managing in Component View, [13-55](#)
 - managing in Page Designer, [13-51](#)
 - names, [13-61](#)
 - template, [11-49](#)
 - Warn on Unsaved Changes, [13-54](#)
 - button template
 - Definition, [11-50](#)
 - Name, [11-50](#)
 - Substitution Strings, [11-50](#)
 - button template substitution strings
 - ##ICON_CSS_CLASSES##, [11-51](#)
 - #BUTTON_ATTRIBUTES#, [11-51](#)
 - #BUTTON_CSS_CLASSES#, [11-51](#)
 - #BUTTON_ID#, [11-51](#)
 - #LABEL#, [11-51](#)
 - #LINK#, [11-51](#)
- ## C
-
- Cache attribute, [6-30](#)
 - caching regions, [12-9](#), [12-22](#)
 - calendar
 - accessing attributes, [10-7](#), [24-65](#)
 - add initialization JavaScript code, [10-9](#)
 - adding Calendar, [10-4](#)
 - adding to new page, [10-4](#)
 - creating, [10-1](#)
 - creating in Page Designer, [10-5](#)
 - creating legacy calendar, [24-61](#)
 - creating Legacy Classic Calendar, [24-62](#)
 - editing attributes, [10-6](#)
 - enabling drag and drop, [10-10](#)
 - including add and edit functions, [10-7](#), [24-66](#)
 - integrating with Google calendar, [16-10](#)
 - managing in Legacy Component View, [24-64](#)
 - supported substitution strings, [24-67](#)
 - viewing sample, [10-3](#)
 - calendar attributes
 - accessing, [10-7](#), [24-65](#)
 - cascading style sheets
 - cascading style sheets (*continued*)
 - about, [11-32](#)
 - referencing for SVG charts, [24-49](#)
 - referencing in page template, [11-81](#)
 - referencing inline (SVG charts), [24-50](#)
 - uploading, [11-81](#)
 - using custom, [11-81](#)
 - Change History report, [6-77](#)
 - Character Value Comparison, [6-35](#)
 - Character Value Comparison Behavior, [6-35](#)
 - chart
 - creating with Create Application Wizard, [6-4](#)
 - chart synchronous update
 - creating refresh dynamic action, [10-30](#), [24-40](#), [24-44](#)
 - editing chart attributes, [24-40](#), [24-44](#)
 - chart syntax, [24-33](#)
 - chart type
 - switching in page designer, [10-29](#)
 - charts
 - about Flash, [24-33](#)
 - about wizards, [24-33](#)
 - adding to a page, [10-27](#), [24-37](#), [24-41](#)
 - adding to new page, [10-25](#)
 - creating Legacy AnyChart Charts, [24-35](#)
 - altering display, [10-29](#), [24-38](#), [24-43](#)
 - asynchronous updates, [24-39](#), [24-44](#)
 - automatic refresh, [10-30](#)
 - automatic refresh attribute
 - enabling, [10-30](#)
 - candlestick syntax, [24-35](#)
 - chart syntax, [24-33](#)
 - creating, [10-23](#)
 - creating with Create Application Wizard, [6-3](#)
 - custom JavaScript, [10-32](#)
 - dial chart syntax, [24-34](#)
 - displaying SVG charts in other languages, [24-50](#)
 - dynamic action, [10-30](#)
 - editing attributes, [10-29](#), [24-38](#), [24-43](#)
 - enabling asynchronous updates, [24-40](#), [24-44](#)
 - Gantt syntax, [24-35](#)
 - HTML chart support, [24-51](#)
 - HTML5 chart about, [24-33](#)
 - legacy, [24-32](#)
 - managing in Component View, [24-41](#)
 - managing in Page Designer, [10-27](#), [24-37](#)
 - migrating SVG to Flash, [24-46](#)
 - monitoring information, [24-39](#), [24-44](#)
 - multiple series syntax, [24-34](#)
 - providing a SQL query, [24-33](#)
 - range syntax, [24-34](#)
 - scatter syntax, [24-34](#)
 - SQL query syntax, [24-33](#)

- charts (*continued*)
 - SVG chart support, [24-45](#)
 - switching to Flash, [24-39](#), [24-43](#)
 - types, [24-33](#)
 - upgrading a SVG chart to HTML5, [24-46](#)
 - upgrading all SVG charts to HTML5, [24-46](#)
 - upgrading SVG charts to HTML5, [24-45](#)
- check box, [A-2](#)
 - creating dependent LOV, [13-25](#)
- classic calendar template
 - comments, [11-52](#)
 - Daily Calendar, [11-53](#)
 - Monthly Calendar, [11-53](#)
 - Name, [11-52](#)
 - Template Subscription, [11-52](#)
 - Weekly Calendar, [11-53](#)
- classic report
 - adding download link, [8-108](#), [8-118](#)
 - changing, [8-102](#), [8-112](#)
 - changing alignment, [8-102](#), [8-112](#)
 - changing column attributes, [8-102](#), [8-112](#)
 - changing column text, [8-102](#), [8-112](#)
 - changing pagination, [8-104](#), [8-114](#)
 - controlling column breaks, [8-111](#), [8-123](#)
 - controlling column display, [8-109](#), [8-122](#)
 - creating, [8-5](#)
 - creating a sum, [8-102](#), [8-112](#)
 - creating updatable columns, [8-109](#), [8-120](#)
 - defining a column as a list of values, [8-110](#), [8-121](#)
 - editing attributes, [8-103](#), [8-112](#)
 - editing report layout, [8-102](#), [8-113](#)
 - exporting, [8-106](#), [8-118](#)
 - hiding columns, [8-102](#), [8-112](#)
 - sorting columns, [8-102](#), [8-112](#)
 - synchronizing column order with region
 - source, [8-123](#), [24-61](#)
- classic report regions
 - configuring print attributes, [8-126](#)
 - printing methods, [8-125](#)
- classic reports
 - about, [8-4](#)
- Cleanup PL/SQL Code, [6-31](#)
- collections, [17-7](#)
- color picker, [A-2](#)
 - configure display style, [16-9](#), [16-10](#)
- columns templates, using JavaScript, [11-80](#)
- Comments attribute, [24-25](#)
- commit points, [2-10](#)
- Compatibility Mode attribute, [6-19](#)
- Component Settings
 - adding sliders, [16-11](#)
 - Color Picker, [16-9](#), [16-10](#)
 - integrating with Google calendar, [16-10](#)
 - interactive report Action menu, [16-11](#)
- Component Settings (*continued*)
 - managing, [16-9](#)
 - switch defaults, [16-11](#)
 - viewing History report, [16-12](#)
- Component View
 - adding a page control, [24-12](#)
 - Branches, [24-8](#)
 - Breadcrumbs, [24-9](#)
 - Buttons, [24-7](#)
 - Computations, [24-8](#)
 - configuring dialog attributes, [24-27](#)
 - Copy icon, [24-11](#)
 - copying page, [24-30](#)
 - Create icon, [24-11](#)
 - Create menu, [24-5](#)
 - Create, Developer Comment, [24-5](#)
 - Create, Page, [24-5](#)
 - Create, Page as Copy, [24-5](#)
 - Create, Shared Components, [24-5](#)
 - Create, Team Development, [24-5](#)
 - deleting application page, [24-31](#)
 - Edit All icon, [24-11](#)
 - Edit All Icon, [24-11](#)
 - Items, [24-7](#)
 - List of Values, [24-9](#)
 - Lists, [24-9](#)
 - marking region titles as translatable, [20-4](#)
 - Navigation Bar, [24-9](#)
 - Next Page button, [24-5](#)
 - Page, [24-7](#)
 - Page Designer button, [24-6](#)
 - Page Selector, [24-5](#)
 - Previous Page button, [24-5](#)
 - Processes, [24-8](#)
 - Regions, [24-7](#)
 - running a page, [24-28](#)
 - Security, [24-9](#)
 - Tabs, [24-9](#)
 - Templates, [24-9](#)
 - Theme, [24-9](#)
 - using, [24-10](#)
 - Utilities, [24-5](#)
 - Utilities, Application Utilities, [24-5](#)
 - Utilities, Attribute Dictionary, [24-5](#)
 - Utilities, Caching, [24-5](#)
 - Utilities, Cross Page Utilities, [24-5](#)
 - Utilities, Delete, [24-5](#)
 - Utilities, Delete Page, [24-5](#)
 - Utilities, Export, [24-5](#)
 - Utilities, History, [24-5](#)
 - Utilities, Page Groups, [24-5](#)
 - viewing field-level help, [1-27](#)
- Component View tab
 - adding buttons, [7-28](#)
 - adding items, [7-28](#)

- Component View tab (*continued*)
 - adding regions, 7-28
- Component View toolbar
 - about, 24-4
 - Create, Team Development, 24-4
 - Edit Page, 24-5
 - Shared Components, 24-4
- Component View, page attributes, 24-12
 - Authentication, 24-19, 24-20
 - Authorization Scheme, 24-19
 - Browser Cache, 24-21
 - Cache Page, 24-23
 - Cache Page Condition, 24-23
 - Cache Timeout, 24-23
 - Comments, 24-25
 - Configuration, 24-23
 - CSS, Inline, 24-17
 - Cursor Focus, 24-15
 - Deep Linking, 24-20
 - Dialog, Attributes, 24-18
 - Dialog, CSS Classes, 24-18
 - Dialog, Height, 24-17
 - Dialog, Maximum Width, 24-17
 - Dialog, Width, 24-17
 - display attributes, 24-14
 - Duplicate Submission, 24-22
 - Error Handling, 24-24
 - Error Handling Function, 24-24
 - Expression 1, 24-23
 - File URLs, 24-16
 - Footer, 24-18, 24-19
 - Form Auto Complete, 24-20
 - Groups, 24-14
 - Header and Footer, 24-18
 - Header Text, 24-18, 24-19
 - Help, 24-25
 - In-line Error Notification Text, 24-24
 - JavaScript, Execute when Page Loads, 24-16
 - JavaScript, File URLs, 24-16
 - JavaScript, Function and Global Variable Declaration, 24-16
 - Name, 24-14
 - Page Alias, 24-14
 - Page CSS Classes, 24-15
 - Page Mode, 24-15
 - Page Template Options, 24-15
 - Read Only, 24-19
 - Rejoin Sessions, 24-22
 - security, 24-19
 - Server Cache, 24-23
 - Title, 24-14
- Component View, Page Processing
 - Branches, 24-8
 - Computations, 24-8
- Component View, Page Processing (*continued*)
 - Processes, 24-8
- Component View, Page Rendering
 - Buttons, 24-7
 - Dynamic Actions, 24-8
 - Items, 24-7
 - Page, 24-7
 - Processes, 24-8
 - Regions, 24-7
- Component View, Shared Components
 - Breadcrumbs, 24-9
 - List of Values, 24-9
 - Lists, 24-9
 - Navigation Bar, 24-9
 - Security, 24-9
 - Tabs, 24-9
 - Templates, 24-9
 - Theme, 24-9
- components
 - attaching an authorization scheme, 19-64
 - controlling access to, 2-12
 - displaying on all pages, 12-3
 - displaying on every page, 12-2
 - translating messages, 20-15, 20-16, 20-35
- components, managing availability, 19-13
- condition type, 2-11
- conditional
 - processing, 2-10
 - rendering, 2-10
- conditions
 - displaying regions, 12-21
 - list of, C-1
 - using, 2-10
 - using with buttons, 13-53, 13-59
- configuration, controlling, 23-40
- confirmation page
 - about creating, 12-17
- controls, controlling access to, 2-12
- Create Application Wizard, 2-1
 - about, 6-2
 - adding a column toggle report, 6-4
 - adding blank page, 6-3
 - adding chart, 6-3, 6-4
 - adding column toggle report, 8-7
 - adding editable interactive grid, 6-3
 - adding form, 6-3, 6-4
 - adding list view report, 6-4
 - adding master detail, 6-3
 - adding reflow report, 6-4
 - adding report, 6-3
 - adding report and form, 6-3, 8-6
 - adding reports, 8-6
 - creating forms, 9-9
 - creating list view and form, 8-7
 - creating list view report, 8-6, 8-7

- Create Application Wizard (*continued*)
 - creating reflow report, 8-7
 - creating reports, 8-9
 - editable interactive grid, 8-6
 - page types, 6-2
 - report options, 8-6
- Create Branch Wizard, 14-25, 14-27
- Create Navigation Bar Entry Wizard, 10-39, 14-21–14-23
- Create Page Computation Wizard, 15-2, 15-5, 17-24
- Create Page Process Wizard, 15-27
- Create Page Wizard, 6-45
 - adding interactive grid, 9-6
 - adding master detail, 9-6
 - breadcrumb, 12-15, 12-17
 - calendar, 12-15, 12-17
 - chart, 12-15, 12-17
 - creating a wizard, 6-46
 - creating access control list, 6-47, 6-48
 - creating blank page, 6-46, 6-47
 - creating calendar, 6-46, 6-48
 - creating chart, 6-46, 6-48
 - creating classic report, 8-8
 - creating column toggle report, 8-9
 - creating Data Load Wizard
 - Data Load Wizard, 17-1
 - creating data loading wizard, 6-46
 - creating feedback page, 6-46, 6-48
 - creating form on a procedure, 9-8
 - creating form on a SQL query, 9-7
 - creating form on procedure, 9-7
 - creating form on SQL Query, 9-8
 - creating form on table, 9-7
 - creating form on Web service, 9-7
 - creating forms, 9-8
 - creating global page, 6-47
 - creating interactive grid, 8-8
 - creating interactive report, 8-8
 - creating legacy page, 6-47, 6-48
 - creating list view, 8-9
 - creating plug-ins, 6-46, 6-47
 - creating reflow report, 8-9
 - creating report and form on table, 8-8
 - creating report and form on Web Service, 9-8
 - creating report on Web service, 8-8, 8-9
 - creating report with a list view on table, 9-7
 - creating report with form on table, 9-6
 - creating reports, 8-11
 - creating single page master detail, 9-7
 - creating tree, 6-46
 - creating with Create Page Wizard, 9-7
 - creating wizard, 6-48
 - Data Load Wizard
 - about pages created, 17-1
- Create Page Wizard (*continued*)
 - Data Load Wizard (*continued*)
 - Data Load Definition, 17-1
 - form and report on Web service, 12-15, 12-17
 - Form and Report on Web Service, 9-7
 - form on a SQL query, 12-14, 12-17
 - form on table, 9-7
 - form on Web service, 12-17
 - form on Web Service, 9-8
 - form options, 9-6
 - list, 12-15
 - map, 12-15
 - plug-ins, 12-15, 12-17
 - report options, 8-7
 - summary page, 12-17
 - tree, 12-15
- Create Region Wizard, 10-21, 12-18, 12-27, 12-28, 14-15, 16-39, 24-41
 - adding column toggle report, 12-16
 - adding reflow report, 12-16
 - classic report, 12-14, 12-16
 - creating breadcrumb, 14-11
 - creating forms, 9-10
 - Form on a Procedure, 12-14, 12-16
 - Form on a Table, 12-14
 - Form on Table with List View, 12-17
 - Help text, 12-15, 12-18
 - interactive report, 12-14
 - list view, 12-16
 - Multiple Static Content, 12-14, 12-16
 - PL/SQL dynamic content, 12-15, 12-17
 - region display selector, 12-15, 12-18
 - Report on Web service result, 12-14, 12-16
 - Static Content, 12-14, 12-16
 - URL, 12-15, 12-18
- Create Validations Wizard, 15-12, 15-15–15-17
- cross page utilities
 - accessing, 6-80
 - Delete Multiple Pages, 6-80
 - Grid Edit of all Pages, 6-80
 - Page Attributes, 6-80
 - Page Locks, 6-80
- cross site-scripting
 - about, 19-21
 - about safe item types, 19-22
 - dynamic output, 19-23
 - escaping values, 19-22
 - HTML region, 19-21
 - protecting dynamic output, 19-23
 - protecting form items, 19-25
 - protecting report regions, 19-24
 - report regions, 19-24
 - restricting characters on items, 19-25
 - static display areas, 19-21

cross site-scripting (*continued*)
 understanding session state protection,
 19-26
 using safe item types, 19-23

CSS classes, SVG charts, 24-46

CSS Finder, 6-55

Custom Authentication
 setting up, 19-50

custom packaged application
 about, 23-8
 about Supporting Objects page, 23-9
 adding access control list, 23-12
 build options, 23-10
 creating supporting objects, 23-9
 defining an upgrade script, 23-14
 deinstallation scripts, 23-12
 deinstalling, 23-15
 deleting installation options, 23-13
 deleting messages, 23-13
 deleting supporting object scripts, 23-13
 how to create, 23-8
 installation scripts, 23-10
 installing supporting objects, 23-13
 messages, 23-10, 23-12
 prerequisites, 23-10
 substitutions strings, 23-10
 testing installation scripts, 23-13
 upgrade messages, 23-11
 upgrade scripts, 23-11
 upgrading, 23-15
 validations, 23-10
 viewing an install summary, 23-16

D

DAD Credentials Verification
 setting up, 19-55

Data Load Wizard
 about creating, 17-1
 adding to application, 17-1
 creating, 17-3
 creating with Create Page Wizard, 17-1
 re-creating pages, 17-5

data loading
 adding, 17-1

Database Account Credentials authentication
 setting up, 19-51

database application
 about, 2-2
 adding a page, 6-44
 adding components, 6-45
 adding pages, 6-45
 assigning to groups, 6-72
 based on queries, 6-5
 based spreadsheet, 6-8

database application (*continued*)
 copying, 6-9
 copying from Application home page, 6-11
 copying with Create Application Wizard, 6-10
 creating groups, 6-72
 creating on tables, 6-5
 deleting, 6-15
 groups, 6-72
 removing from groups, 6-73
 running, 2-7

database application groups
 assigning pages, 6-72
 creating, 6-72
 removing, 6-73
 removing pages, 6-73

database application page, 24-31
 adding, 6-44
 assigning to groups, 6-63
 copying in Component View, 24-30
 copying in Page Designer, 7-39
 creating from Application home, 6-48
 deleting, 7-40
 deleting a range of, 24-32
 deleting in Component View, 24-31
 deleting in Page Designer, 7-40
 deleting multiple, 7-41, 24-31
 deleting range of, 7-41
 determining lock status, 6-64
 grouping, 6-61
 locking, 6-64
 managing, 6-44
 preventing conflicts, 6-64
 removing from groups, 6-63
 running, 2-7
 unlocking, 6-64, 6-66

database application page groups, 6-62
 assigning pages, 6-63
 creating, 6-62
 editing group definition, 6-62
 removing, 6-63
 removing pages, 6-63
 viewing, 6-61

database link, 17-30
 about, 17-31
 creating, 17-31
 viewing existing database links, 17-31

Database Object Dependencies report, 6-78

Database Session attribute, 6-31

database users
 granted privileges, D-1
 granting privileges, D-1

date picker, A-3

DBMS_APPLICATION_INFO, 22-3

Debug report
 viewing, 22-7

- debugging, [21-1](#), [22-1](#)
 - checking for errors, [22-1](#)
 - enabling and disabling, [22-5](#)
 - isolating a problem, [22-8](#)
 - running with application, [22-5](#)
 - SQL queries, [21-3](#)
 - SQL tracing, [22-3](#)
 - using debug mode, [22-4](#)
 - using f?p to access, [22-6](#)
 - viewing reports at runtime, [22-7](#)
 - viewing reports in development mode, [22-6](#)
- Debugging attribute, [6-19](#)
- deep linking, [19-46](#)
- Deep Linking attribute, [6-27](#)
- Define Authorization Scheme, [6-25](#)
- defining, [17-10](#)
- deleting export files, [23-38](#)
- demonstration application
 - about Sample Database Application, [3-8](#)
- deployment, [23-6](#)
 - creating supporting objects, [23-9](#)
 - exporting a page, [23-24](#)
 - exporting a Websheet, [23-25](#)
 - exporting application definition, [23-19](#)
 - exporting feedback, [23-29](#)
 - exporting script files, [23-30](#)
 - exporting static files, [23-25](#)
 - exporting themes, [23-26](#)
 - exporting User Interface Defaults, [23-27](#)
 - exporting workspaces, [23-23](#)
 - importing application from SQL*Plus, [23-38](#)
 - importing export files, [23-30](#)
 - importing files, [23-30](#)
 - installing application export, [23-36](#)
 - installing files, [23-37](#)
 - managing database objects, [23-7](#)
 - of an application, [23-4](#)
 - options, [23-5](#)
 - publishing the URL, [23-38](#)
 - publishing the Websheet application URL, [23-39](#)
 - understanding the process, [23-4](#)
- developer comments
 - adding in Page Designer, [6-68](#)
 - adding to an application, [6-69](#)
 - adding to application, [6-69](#)
 - deleting, [6-69](#)
 - editing, [6-69](#)
 - viewing, [6-69](#)
 - viewing reports, [6-70](#)
- Developer Comments calendar, [6-71](#)
- Developer Navigation tools
 - about, [1-6](#)
 - Account menu, [1-10](#)
 - Administration menu, [1-8](#)
- Developer Navigation tools (*continued*)
 - global search, [1-7](#)
 - Help menu, [1-9](#)
- Developer toolbar
 - Session, [2-14](#)
- development process
 - managing, [4-1](#)
- development process, managing, [4-1](#)
- dial chart syntax, [24-34](#)
- dialog
 - configuring attributes, [24-27](#)
 - configuring attributes in Page Designer, [7-35](#)
- dialog page template, [6-50](#)
- dialog page template attributes, [11-67](#)
- dialog pages, [6-49](#)
 - About branching, [6-53](#)
 - about supported, [6-49](#)
 - creating a dialog page template, [6-50](#)
 - defining, [6-50](#), [6-51](#)
 - generating a Cancel dynamic action, [6-52](#)
 - generating a close dialog page process, [6-52](#)
 - handling refresh actions, [6-53](#)
 - Modal Dialog, [6-49](#)
 - non-modal dialog, [6-49](#)
 - required template attributes, [11-67](#)
 - using dialog page template, [6-49](#)
- DML
 - about locking, [17-32](#)
 - APEX_DML_LOCK_WAIT_TIME, [17-32](#)
 - FSP_DML_LOCK_ROW, [17-32](#)
- Document Direction attribute, [6-34](#)
- documentation
 - about, [1-25](#)
 - accessing, [1-25](#)
- documents
 - downloading, [1-26](#)
- dynamic action
 - calendars, [10-11](#)
- Dynamic Actions
 - adding Actions, [13-48](#)
 - creating, [13-40](#), [13-45](#)
 - debugging, [13-50](#)
 - deleting, [13-44](#)
 - editing, [13-43](#), [13-47](#)
 - frequency and scope, [13-44](#), [13-49](#)
 - managing, [13-36](#)
 - managing in Component View, [13-45](#)
 - managing in Page Designer, [13-40](#)
 - understanding, [13-36](#)
- dynamic list
 - about creating, [16-35](#)
 - creating from scratch, [16-37](#)
 - syntax, [16-36](#)
- dynamic translation, [20-58](#)

E

- Edit All icon, [24-11](#)
- Edit Application Definition
 - Allow Feedback, [6-19](#)
 - Application Alias, [6-18](#)
 - Application Groups, [6-18](#)
 - Build Status, [6-20](#)
 - Compatibility Mode, [6-19](#)
 - Debugging, [6-19](#)
 - Email from Address, [6-19](#)
 - Error Handling, Default Error Display
 - Location, [6-21](#)
 - Error Handling, Error Handling Function, [6-22](#)
 - Global Notifications, [6-22](#)
 - Logging, [6-18](#)
 - Message for unavailable application, [6-20](#)
 - Name attribute, [6-17](#)
 - Proxy Server, [6-19](#)
 - restrict access, [6-21](#)
 - Status, [6-20](#)
 - Version, [6-18](#)
- Edit Application Definition page
 - about, [6-17](#)
 - accessing, [6-16](#)
- Edit Application page
 - Database Session, [6-31](#)
- Edit Globalization Attributes, [6-32](#)
 - Application Date Format, [6-34](#)
 - Application Date Time Format, [6-34](#)
 - Application Language Derived From, [6-33](#)
 - Application Primary Language, [6-33](#)
 - Application Timestamp Format, [6-34](#)
 - Application Timestamp Time Zone Format, [6-34](#)
 - Automatic CSV Encoding, [6-35](#)
 - Automatic Time Zone, [6-35](#)
 - Character Value Comparison, [6-35](#)
 - Character Value Comparison Behavior, [6-35](#)
 - Document Direction, [6-34](#)
- Edit Security Attributes, [6-23](#)
 - accessing, [6-24](#)
 - Authentication, [6-24](#)
 - Authorization, [6-25](#)
 - Session Management, [6-26](#)
 - Session State Protection, [6-29](#)
- editable interactive grid, [9-1](#)
 - creating with Create Application Wizard, [6-3](#)
- email
 - configuring for instance, [18-2](#)
 - email log, [18-1](#)
 - email queue, [18-1](#)
 - enabling, [19-11](#)
 - managing, [18-1](#)
 - sending with APEX_MAIL package, [18-3](#)

- Email from Address attribute, [6-19](#)
- Embed in Frames attribute, [6-30](#)
- embedded PL/SQL gateway
 - security practices, [19-4](#)
- engine, [1-3](#)
- error message
 - creating for a page process, [15-25](#), [15-30](#)
 - creating for application processes, [15-36](#)
- errors, handling, [6-21](#), [24-24](#)
- Expire Bookmarks attribute, [6-29](#)
- export
 - application, [23-19](#)
 - application attributes, [23-21](#)
 - build option status, [23-21](#)
 - database application, [23-16](#)
 - database application components, [23-16](#)
 - feedback, [23-29](#)
 - managing database objects, [23-7](#)
 - page, [23-24](#)
 - page components, [23-21](#)
 - plug-in, [23-26](#)
 - script file, [23-30](#)
 - shared components, [23-21](#)
 - static files, [23-25](#)
 - text for translations, [20-9](#)
 - themes, [23-26](#)
 - User Interface Defaults, [23-27](#)
 - Websheet, [23-25](#)
 - workspaces, [23-23](#)
- Export Repository, [23-36–23-38](#)
 - accessing, [23-36](#)
- Export/Import icon, [1-19](#)
- exported application, importing, [23-30](#)

F

- f?p syntax, [2-17](#)
- feature reports
 - approval statuses, [4-9](#)
 - Calendar page, [4-8](#)
 - Change History, [4-8](#)
 - Feature Utilities, [4-10](#)
 - Focus Areas, [4-9](#)
 - Owners, [4-9](#)
 - Progress Log, [4-8](#)
 - Tree page, [4-7](#)
- feature utilities
 - assigning milestones, [4-27](#)
 - change milestones for overdue features, [4-27](#)
 - set feature due date to milestone, [4-27](#)
- Feature utilities
 - push due date of open features, [4-27](#)
- features
 - attaching files, [4-5](#)
 - creating, [4-4](#)

features (*continued*)

- tracking, [4-3](#)
- updating, [4-5](#)
- viewing, [4-4](#)
- viewing reports, [4-7](#)

feedback

- adding a page, [4-21](#)
- Feedback page, [4-21](#)
- reviewing comments, [4-22](#)
- using, [4-22](#)
- viewing, [4-21](#)
- viewing reports, [4-23](#)

feedback reports

- Calendar, [4-23](#)
- Tag Cloud, [4-3](#)

field-level Help, [1-27](#)

- viewing in Component View, [1-27](#)
- viewing in Page Designer, [1-28](#)

file upload table

- security risks, [19-39](#)

Find icon, [6-55](#), [6-56](#), [24-5](#)

Flash chart

- adding to new page, [24-35](#)
- candlestick syntax, [24-35](#)
- Gantt syntax, [24-35](#)
- multiple series syntax, [24-34](#)
- range syntax, [24-34](#)
- saving locally, [24-52](#)
- scatter syntax, [24-34](#)
- using custom XML, [24-52](#)

focus areas

- editing, [4-27](#)

footer, substitution strings, [12-8](#), [12-21](#)

form

- creating with Create App Wizard, [6-3](#)
- creating with Create Application Wizard, [6-4](#)

Form

- legacy master detail form, [24-53](#)

form on a table, [9-2](#)

forms

- Automatic Row Processing (DML) process, [9-11](#)
- creating a tabular form, [24-56](#)
- creating manually, [9-11](#)
- creating single page master detail, [9-19](#)
- creating two page master detail, [9-21](#)
- creating using Create Page Wizard, [9-8](#)
- creating with Create Application Wizard, [9-9](#)
- creating with Create Region Wizard, [9-10](#)
- editable interactive grid, [9-1](#)
- form on a table, [9-2](#)
- managing master detail, [9-14](#)
- master detail, [9-15](#)
- populating, [9-13](#)
- understanding processing, [9-11](#)

forms (*continued*)

- understanding types, [9-1](#)
- validating input, [9-22](#)

full development environment, [1-3](#)

functions, searching for, [6-59](#)

G

Gallery

- about, [7-13](#)
- using, [7-25–7-27](#)
- viewing legacy components, [24-68](#)

Gallery menu

- Show Legacy Components, [24-68](#)

Global Notifications attribute, [6-22](#)

Global page, [12-3](#)

- creating, [12-3](#)
- navigating to, [12-4](#)

global template option group, [11-36](#)

- creating, [11-36](#)
- editing, [11-37](#)

global template options

- creating, [11-39](#)
- editing, [11-39](#)
- managing, [11-37](#)

Global Templates, [11-15](#)

global variable, creating, [16-23](#)

globalization

- attributes, [20-5](#)
- determining application language, [6-32](#)
- format masks, [20-6](#)
- understanding, [20-1](#)

graphics

- managing, [12-31](#)

grid messages requiring translation

- interactive grid, [20-17](#)

groups

- of applications, [6-71](#)
- of database application pages, [6-61](#)

H

Help

- accessing, [1-25](#)
- creating, [10-33](#)
- creating navigation bar entry, [10-39](#)
- editing, [10-38](#)
- searching, [1-25](#)
- using, [1-25](#)

Help menu, [1-9](#)

Help text

- defining for item, [10-37](#)
- defining for page, [10-37](#)
- editing in bulk, [10-38](#)
- editing items, [10-38](#)

- Help text (*continued*)
 - seeding, [10-38](#)
 - hidden item, [A-6](#)
 - HTML
 - generated using custom PL/SQL, [12-31](#)
 - not handled by Application Express, [12-31](#)
 - HTML chart
 - upgrading all HTML charts to HTML5, [24-52](#)
 - upgrading to HTML5, [24-51](#)
 - HTML Escaping Mode attribute, [6-30](#)
 - HTML password form element, [A-8](#)
 - HTML text area, [A-11](#)
 - HTML5 chart
 - about, [24-33](#)
 - adding to new page, [24-35](#)
 - multiple series syntax, [24-34](#)
 - switching to Flash chart, [24-39](#), [24-43](#)
 - using custom XML, [24-52](#)
 - HTTP Header Variable authentication
 - setting up, [19-52](#)
 - HTTP Response Headers attribute, [6-31](#)
-
- I
- Image Finder, [6-55](#)
 - Image Prefix attribute, [6-38](#)
 - Image Repository, [12-31](#)
 - image, displaying from URL, [A-3](#)
 - images
 - managing, [12-31](#)
 - searching for, [6-60](#)
 - viewing available, [6-60](#)
 - Images Finder, [6-60](#)
 - implementing
 - plug-ins, [18-3](#)
 - import
 - feedback, [23-35](#)
 - Websheet, [23-32](#)
 - importing
 - application, [23-31](#)
 - application using SQL*Plus, [23-38](#)
 - export files, [23-30](#)
 - page, [23-31](#)
 - plug-in, [23-33](#)
 - themes, [23-33](#)
 - User Interface Defaults, [23-34](#)
 - Initialization PL/SQL Code, [6-31](#)
 - injection attacks, restricting characters on items, [19-25](#)
 - installation
 - enabling network services, [19-14](#)
 - installing
 - application files, [23-36](#)
 - export files, [23-37](#)
 - installing application export, [23-36](#)
 - installing export files, [23-37](#)
 - instance proxy, specifying, [19-6](#)
 - instance settings
 - configuring, [19-10](#)
 - configuring email, [18-2](#)
 - interactive grid
 - creating, [8-5](#)
 - creating with Create Page Wizard, [9-6](#)
 - managing, [8-13](#)
 - message requiring translation, [20-17](#)
 - interactive grid messages, translating, [20-17](#)
 - interactive grids
 - about, [8-1](#)
 - controlling pagination, [8-22](#)
 - creating a column link, [8-32](#)
 - creating master detail form, [8-15](#)
 - customizing in runtime, [8-33](#)
 - customizing toolbar, [8-23](#)
 - defining a column as list of values, [8-31](#)
 - deleting a report, [8-37](#)
 - displaying conditionally, [8-19](#)
 - editing Attributes, [8-20](#)
 - editing column attributes, [8-30](#)
 - enabling Detail view, [8-28](#)
 - enabling icon view, [8-26](#)
 - enabling saving public reports, [8-24](#)
 - making editable, [8-21](#)
 - managing column attributes, [8-29](#)
 - managing region attributes, [8-14](#), [8-20](#)
 - renaming saved report, [8-37](#)
 - resetting a report, [8-38](#)
 - restricting saving of Public reports, [8-25](#)
 - saving, [8-33](#)
 - saving and authentication, [8-35](#)
 - saving configuration dependencies, [8-35](#)
 - saving Primary report, [8-36](#)
 - saving private report, [8-35](#)
 - saving public report, [8-36](#)
 - user save options, [8-34](#)
 - interactive report filters, [8-95](#)
 - editing, [8-95](#)
 - enabling and disabling, [8-95](#)
 - removing, [8-95](#)
 - showing and hiding details, [8-95](#)
 - interactive report messages, translating, [20-41](#)
 - interactive report regions
 - configuring print attributes, [8-128](#)
 - interactive reports
 - about, [8-2](#)
 - about link columns, [8-52](#), [8-76](#)
 - about Search bar, [8-93](#)
 - Actions menu, [8-94](#)
 - configuring structure, [16-11](#)
 - adding download option, [8-47](#), [8-73](#)
 - advanced attributes, [8-62](#), [8-84](#)

- interactive reports (*continued*)
 - Attributes, [8-42](#)
 - Attributes page, [8-65](#)
 - changing pagination, [8-43](#), [8-68](#)
 - column attributes, [8-40](#), [8-67](#)
 - configuring Actions menu structure, [16-11](#)
 - creating, [8-5](#)
 - creating a column link, [8-56](#), [8-79](#)
 - creating groups, [8-68](#)
 - creating with Create Region Wizard, [12-14](#)
 - customizing in runtime, [8-92](#)
 - defining a column as a list of values, [8-57](#), [8-81](#)
 - deleting as a developer, [8-86](#)
 - deleting private report, [8-101](#)
 - deleting public report, [8-101](#)
 - displaying conditionally, [8-57](#), [8-80](#)
 - editing as a developer, [8-38](#), [8-64](#)
 - Email From Address, [8-63](#), [8-85](#)
 - enabling Icon view, [8-59](#), [8-82](#)
 - exporting, [8-47](#), [8-73](#)
 - filtering on conditional links, [8-41](#)
 - formatting columns, [8-58](#), [8-81](#)
 - invalid linking examples, [8-90](#)
 - linking restrictions, [8-89](#)
 - linking to, [8-87](#)
 - linking with clear cache syntax, [8-89](#)
 - linking with name and value syntax, [8-88](#)
 - linking with Request syntax, [8-88](#)
 - managing as a developer, [8-86](#)
 - managing saved reports, [6-81](#)
 - managing subscriptions, [6-81](#)
 - message requiring translation, [20-41](#)
 - Region Alias, [8-63](#), [8-84](#)
 - renaming private report, [8-100](#)
 - renaming public report, [8-100](#)
 - Report ID Item, [8-63](#), [8-84](#)
 - saving, [8-96](#)
 - saving alternative, [8-99](#)
 - saving Alternative report, [8-37](#)
 - saving as private, [8-98](#)
 - saving as public, [8-98](#)
 - saving primary default, [8-99](#)
 - user save options, [8-96](#)
 - using filters, [8-95](#)
 - valid linking examples, [8-90](#)
 - viewing as a developer, [8-86](#)
 - viewing saved report activity, [8-87](#)
 - internal messages, requiring translation, [20-37](#)
 - IP address, restricting access, [19-5](#)
 - isolating workspaces, about, [19-8](#)
 - item
 - naming conventions, [13-2](#)
 - referencing values, [13-2](#)
 - searching for, [6-56](#)
 - item (*continued*)
 - using format masks when translating, [20-6](#)
 - Item Finder, [6-55](#), [6-56](#)
 - item help
 - editing multiple topics, [10-38](#)
 - item Help
 - editing, [10-38](#)
 - viewing, [1-27](#)
 - item type
 - check box, [A-2](#)
 - Color Picker, [A-2](#)
 - Date Picker, [A-3](#)
 - Display Image, [A-3](#)
 - Display Only, [A-4](#)
 - File Browse, [A-5](#)
 - Hidden, [A-6](#)
 - List Manager, [A-6](#)
 - Number Field, [A-7](#)
 - Password, [A-8](#)
 - Percent Graph, [A-8](#)
 - Popup LOV, [A-9](#)
 - Radio, [A-10](#)
 - Rich Text Editor, [A-11](#)
 - Select list, [A-12](#)
 - Shuttle, [A-13](#)
 - Slider, [A-14](#)
 - Switch, [A-13](#)
 - Text Area, [A-17](#)
 - Text Field, [A-15](#)
 - Text Field with autocomplete, [A-16](#)
- ## J
-
- JavaScript, [11-66](#)
 - adding using Component View, [24-25](#)
 - adding using Page Designer, [7-33](#)
 - calling from button, [13-60](#)
 - in row templates, [11-80](#)
 - including in .js file, [11-33](#)
 - including in page template, [11-33](#)
 - incorporating, [13-71](#)
 - libraries, [11-66](#)
 - referencing items, [13-4](#)
 - referencing jQuery library, [18-16](#)
 - setting focus on item, [24-15](#)
 - using a dynamic action, [13-50](#)
 - using in a plug-in, [18-18](#)
 - using jQuery, [18-17](#)
 - JavaScript Finder, [6-55](#)
 - jQuery
 - support, [18-13](#)
 - using in JavaScript, [18-17](#)
 - jQuery Mobile, [6-11](#)
 - support, [6-11](#)
 - jQuery UI

jQuery UI (*continued*)
 accessing other jQuery UI components,
 18-16
 available features, 18-15
 support, 18-13
 jsTree, support, 13-62

L

label template

#CURRENT_ITEM_HELP_TEXT#, 11-55
 creating page item Help, 11-55
 Definition, 11-54
 editing, 11-53
 Error Display, 11-54
 Field Container, 11-55
 Name, 11-53
 Substitution Strings, 11-55

language

defining for application, 20-5
 multibyte, 20-7
 preference, 20-2

layout

about region attributes, 12-19
 controlling, 11-1

LDAP Directory Verification

setting up, 19-53

Legacy AnyChart Charts, 24-35

legacy calendar, 24-61

enabling drag and drop, 24-67

legacy calendar template, 11-51

Legacy Classic Calendar, 24-62

legacy components

viewing in Gallery, 24-68

limits

forms, B-1
 interactive reports, B-1
 item names, B-1
 tabular forms, B-1
 validations, B-1

Link Column

about, 8-52, 8-76
 excluding, 8-55, 8-78
 linking a single row view, 8-52, 8-77
 linking to a page, 8-53, 8-77
 linking to URL, 8-54, 8-78

linking

deep, 19-46

list

about creating, 16-27
 about dynamic, 16-35
 about static, 16-29
 adding to a page, 16-39
 Conditional Entries report, 16-41
 copying, 16-38

list (*continued*)

copying between applications, 16-39
 copying from current application, 16-38
 creating hierarchical, 11-56
 editing, 16-40
 editing attributes, 16-40
 History report, 16-41
 Lists page, 16-40
 reports, 16-41
 types, 16-27
 Unused report, 16-41
 Utilization report, 16-41

list manager, creating dependent LOV, 13-25

list of values, 13-25

bulk edit of static LOVs, 16-26
 Conditional Entries report, 16-26
 creating, 16-24
 creating at application-level, 16-25
 Edit Null Text report, 16-26
 editing, 16-25
 Grid Edit report, 16-26
 History report, 16-26
 Locally Defined report, 16-26
 named, 16-25
 referencing session state, 16-26
 reports, 16-26
 shared, 16-25
 static, 16-25
 Subscription report, 16-26
 translating, 20-58
 Utilization report, 16-26

list reports

Conditional Entries, 16-41
 History, 16-41
 Unused, 16-41
 Utilization, 16-41

list template

After List Entry, 11-59
 After Sub List Entry, 11-59
 Before List Entry, 11-57
 Before Sub List Entry, 11-58
 editing, 11-56
 Name, 11-57
 Sub List Entry, 11-58
 Substitution Strings, 11-59, 11-64
 Template Definition, 11-57
 Template Definitions for First Entry, 11-58

List View

adding with Create Region Wizard, 12-16

list view report

creating with Create Application Wizard, 6-4

lists of terms, A-6

Lists page, 16-40

Lists Wizard, 16-29, 16-35

live template options

live template options (*continued*)

using, [11-40](#)

locks

determining lock status, [6-64](#)

for database application page, [6-64](#)

Logging attribute, [6-18](#)

login controls, creating, [19-5](#)

login page

creating, [19-59](#)

logo

defining, [6-39](#)

Logo attribute, [6-38](#)

logs

managing, [21-4](#)

performance, [21-3](#)

M

Manage Session State Protection attribute, [6-29](#)

maps

about map chart support, [10-13](#)

about wizards, [10-13](#)

adding to a a page, [10-17](#), [10-21](#)

adding to new page, [10-16](#)

creating, [10-12](#)

creating SQL queries, [10-13](#)

editing map attributes, [10-18](#), [10-22](#)

enabling asynchronous updates, [10-23](#)

enabling automatic updates (Page Designer),
[10-20](#)

managing in Component View, [10-20](#)

managing in Page Designer, [10-17](#)

providing a SQL query, [10-13](#)

using custom XML, [10-19](#), [10-22](#)

master application, creating, [11-25](#)

master detail

creating single page, [9-19](#), [9-21](#)

creating with Create Application Wizard, [6-3](#),
[9-6](#)

creating with Create Page Wizard, [9-19](#), [9-21](#)

managing, [9-14](#)

master detail form, about, [9-15](#)

master detail forms

about, [9-3](#)

master detail, single page

about, [9-3](#)

master detail, two page

about, [9-3](#)

Maximum Session Idle Time in Second attribute,
[6-28](#)

Maximum Session Length in Seconds attribute,
[6-28](#)

menu, [1-7](#), [1-10](#)

Message for unavailable application, [6-20](#)

messages

messages (*continued*)

creating for translation, [20-36](#)

editing text messages, [20-36](#)

in PL/SQL procedures, [20-16](#)

in reports

translating, [20-35](#)

translating, [20-15](#), [20-16](#)

translating internal, [20-35](#)

messages in reports, [20-35](#)

messages requiring translation

interactive reports, [20-41](#)

internal, [20-37](#)

milestone reports

By Owner report, [4-12](#)

Feature by Milestone report, [4-12](#)

Milestone Calendar, [4-12](#)

milestones

creating, [4-11](#)

tracking, [4-10](#)

updating, [4-11](#)

viewing, [4-10](#)

viewing reports, [4-11](#)

mobile applications, [6-11](#)

about designing, [2-5](#)

creating, [6-11](#), [6-12](#)

jQuery Mobile, [6-11](#)

mobile design, [2-6](#)

responsive design, [2-6](#)

Mobile Theme - 51, [11-2](#)

monitor activity

Calendar Reports, [21-5](#)

Developer Activity, [21-6](#)

Environment Reports, [21-5](#)

Login Attempts, [21-5](#)

Logs, [21-5](#)

Page Views, [21-5](#)

Service Requests, [21-5](#)

Workspace Purge, [21-5](#)

multiple select list item, [13-4](#)

handling return values, [13-5](#)

using APEX_UTIL.STRING_TO_TABLE,
[13-5](#)

N

navigation

branch, [14-25](#)

breadcrumb, [14-8](#)

list, [16-27](#)

parent tabs, [14-2](#)

standard tabs, [14-2](#)

tab, [14-2](#)

navigation bar, [14-19](#)

navigation bar entry, [14-19](#)

about, [14-19](#)

navigation bar entry (*continued*)
 copying, [14-23](#)
 creating, [14-20](#)
 creating for feedback, [14-22](#)
 creating from scratch, [14-20](#)
 editing, [14-23](#)
 editing multiple, [14-24](#)
 History report, [14-24](#)
 reports, [14-24](#)
 Subscription report, [14-24](#)

navigation bar entry reports
 history, [14-24](#)
 subscription, [14-24](#)

network services
 enabling, [19-14](#)
 granting connect privileges, [19-15](#)
 granting connect privileges, Oracle Database
 12c, [19-17](#)
 invalid ACL error, [19-17](#)

news entries, [1-5](#)
 adding, [4-28](#)
 deleting, [4-28](#)
 editing, [4-28](#)

number field, [A-7](#)

NV function, [2-21](#)

O

online Help, [1-25](#)
 searching, [1-25](#)

Open Door Credentials
 setting up, [19-56](#)

Oracle Application Express
 about, [1-2](#)
 architecture, [1-2](#)
 component limits, [B-1](#)
 signing in, [1-1](#)
 user roles, [19-2](#)

Oracle Application Express administrator, [19-2](#)

Oracle Application Express Advisor
 about, [22-1](#)
 checking application integrity, [22-1](#)
 running on a page, [22-2](#)
 running on application, [22-1](#)

Oracle Application Express Engine, [1-3](#)

Oracle BI Publisher, integrating with, [19-14](#)

Oracle Help Center
 searching, [1-25](#), [1-26](#)

Oracle HTTP Server
 about security, [19-3](#)

output escaping
 substitution strings, [2-27](#), [11-33](#)

P

packaged application
 changing authentication, [3-6](#)
 installing, [3-4](#), [3-5](#)
 managing, [3-2](#)
 productivity application, [3-1](#)
 removing, [3-6](#)
 sample application, [3-1](#)
 updating, [3-7](#)

Packaged Applications
 about, [3-3](#)

Packaged Applications page, [3-3](#)
 accessing, [3-3](#)

Packaged Apps Administration
 making service requests, [3-31](#)
 managing saved interactive reports, [3-33](#)
 managing subscriptions, [3-34](#)
 Request Storage, [3-31](#)
 Request Termination, [3-31](#)
 setting workspace preferences, [3-32](#)

Packaged Apps Only workspace, [3-35](#)
 accessing, [3-36](#)
 account expiration, [3-32](#)
 Dashboard, [3-36](#)
 definition, [3-35](#)
 end user account lifetime, [3-32](#)
 locking, [3-32](#)
 managing users, [3-33](#)
 maximum login failures, [3-32](#)
 Packaged App Administration, [3-31](#), [3-36](#)
 Packaged App Gallery, [3-36](#), [3-37](#)
 requesting, [3-35](#)
 viewing activity reports, [3-34](#)
 viewing page views, [3-34](#)
 viewing top users, [3-35](#)

packages, searching for, [6-59](#)

page attributes
 editing in Page Designer, [7-31](#)
 viewing in Component View, [24-13](#)
 viewing in Page Designer, [7-32](#)
 Warn on Unsaved Changes, [7-36](#)

page components, controlling access, [19-44](#)

page computation
 computation point, [15-6](#)
 creating, [15-2](#), [15-5](#)
 defining computation point, [15-4](#), [15-6](#)
 defining computation source, [15-4](#), [15-6](#)
 editing attributes, [15-3](#), [15-5](#)
 making a computation conditional, [15-4](#), [15-6](#)
 Managing in Component View, [15-4](#)
 managing in Page Designer, [15-2](#)
 page submission, [7-30](#)
 pre-rendering, [7-30](#)
 syntax, [15-6](#)

- page control
 - adding from Component View, [24-12](#)
- Page Designer
 - accessing, [7-22](#)
 - accessing the Property Editor, [7-24](#)
 - adding a report, dragging and dropping, [8-11](#)
 - adding buttons, [7-25–7-27](#)
 - adding items, [7-25–7-27](#)
 - adding regions, [7-25–7-27](#)
 - Code Editor, [7-20](#)
 - Component View tab, [7-14](#)
 - configuring dialog attributes, [7-35](#)
 - copying buttons, [7-29](#)
 - copying items, [7-29](#)
 - copying page, [7-39](#)
 - copying regions, [7-29](#)
 - deleting application page, [7-40](#)
 - editing page attributes, [7-32](#)
 - Gallery, [7-13](#)
 - Help, [7-17](#)
 - Layout, [7-11](#)
 - managing page attributes, [7-31](#)
 - marking region titles as translatable, [20-4](#)
 - Messages tab, [7-15](#)
 - Page Search tab, [7-16](#)
 - Property Editor, [7-18](#)
 - Reset Layout, [7-23](#)
 - Right Pane, [7-18](#)
 - running a page, [7-37](#)
 - Runtime Developer toolbar, [7-38](#), [24-29](#)
 - searching for pages, [7-23](#)
 - Three Pane mode, [7-23](#)
 - Tree pane, [7-7](#)
 - Two Pane mode, [7-23](#)
 - using, [7-21](#)
 - viewing field-level help, [1-28](#)
- Page Designer Toolbar, [7-3](#)
 - Create Form Region, [7-5](#)
 - Create menu, [7-3](#)
 - Create, Create Breadcrumb Region, [7-5](#)
 - Create, Developer Comment, [7-3](#)
 - Create, Page, [7-3](#)
 - Create, Page as Copy, [7-3](#)
 - Create, Page Group, [7-3](#)
 - Create, Shared Components, [7-3](#)
 - Create, Team Development, [7-3](#)
 - Developer Comment, Bug, or To Do button, [7-6](#)
 - Go To Page, [7-3](#)
 - Page Lock, [7-3](#)
 - Page Unlock, [7-3](#)
 - Redo, [7-3](#)
 - Save and Run Page button, [7-3](#)
 - Save button, [7-3](#)
 - Settings menu, [7-6](#)
- Page Designer Toolbar (*continued*)
 - Settings, Bugs, [7-6](#)
 - Settings, Features, [7-3](#)
 - Settings, Feedback Entries, [7-6](#)
 - Settings, To Dos, [7-3](#)
 - Settings, Two Pane Mode, [7-3](#)
 - Shared Components, [7-3](#)
 - Undo, [7-3](#)
 - Utilities, [7-3](#)
 - Utilities, Application Utilities, [7-3](#)
 - Utilities, Attribute Dictionary, [7-3](#)
 - Utilities, Delete, [7-3](#)
 - Utilities, Delete Page, [7-3](#)
 - Utilities, Export, [7-3](#)
 - Utilities, History, [7-3](#)
 - Utilities, Page Groups, [7-3](#)
 - Utilities, Upgrade Application, [7-3](#), [24-5](#)
- Page Designer, page attributes
 - Dialog, Attributes, [24-28](#)
 - Dialog, CSS Classes, [24-28](#)
 - Dialog, Height, [24-27](#)
 - Dialog, Maximum Width, [24-28](#)
 - Dialog, Width, [24-27](#)
 - Dialog: Attributes, [7-36](#)
 - Dialog: Chained, [7-36](#)
 - Dialog: CSS Classes, [7-36](#)
 - Dialog: Height, [7-36](#)
 - Dialog: Maximum Width, [7-36](#)
 - Dialog: Width, [7-36](#)
- Page Finder, [6-55](#)
- page item
 - editing display on a page, [13-33](#)
 - types, [A-1](#)
- page layout, about, [12-1](#)
- Page Mode
 - Modal Dialog, [24-14](#)
 - Non-modal Dialog, [24-14](#)
 - Normal, [24-14](#)
- page processes
 - changing processing points, [15-25](#), [15-30](#)
 - changing source, [15-25](#), [15-30](#)
 - creating, [15-23](#), [15-27](#)
 - creating error message, [15-25](#), [15-30](#)
 - creating for a tabular form, [15-28](#)
 - creating success message, [15-25](#), [15-30](#)
 - editing attributes, [15-24](#), [15-29](#)
 - making conditional, [15-26](#), [15-31](#)
 - page submission, [7-30](#)
 - pre-rendering, [7-30](#)
 - When Button Pressed attribute, [2-44](#)
- page processing, [2-10](#)
 - understanding, [2-9](#)
- page rendering, [2-10](#)
 - understanding, [2-9](#)
- Page Selector, [7-23](#)

- page specific utilities, [6-79](#)
 - Branch Utilities, [6-79](#)
 - Button Utilities, [6-79](#)
 - Computations Utilities, [6-79](#)
 - Cross Page Utilities, [6-79](#)
 - Dynamic Action Utilities, [6-79](#)
 - Item Utilities, [6-79](#)
 - Page Groups, [6-79](#)
 - Process Utilities, [6-79](#)
 - Region Utilities, [6-79](#)
 - Validation Utilities, [6-79](#)
- page template
 - Body, [11-60](#)
 - Cascading Style Sheet, [11-61](#)
 - Definition, [11-60](#)
 - Dialog, [11-63](#)
 - Display Points, [11-63](#)
 - editing, [11-59](#)
 - Error Page Template Control, [11-63](#)
 - Footer, [11-60](#)
 - Header, [11-60](#)
 - Image Based Tab Attributes, [11-62](#)
 - JavaScript, [11-61](#)
 - JavaScript, Dialog Cancel Code, [11-61](#)
 - JavaScript, Dialog Closure Code, [11-61](#)
 - JavaScript, Dialog Initialization Code, [11-61](#)
 - JavaScript, Execute when Page Loads, [11-61](#)
 - JavaScript, File URLs, [11-61](#)
 - JavaScript, Function and Global Variable Declaration, [11-61](#)
 - Layout, [11-62](#)
 - Name, [11-60](#)
 - selecting, [11-34](#)
 - specifying defaults, [11-34](#)
 - substitution strings, [11-59](#)
 - Subtemplate, [11-62](#)
- page template substitution strings, [11-59](#)
 - #APEX_CSS#, [11-65](#)
 - #APP_VERSION#, [11-64–11-66](#)
 - #APPLICATION_CSS#, [11-65](#)
 - #APPLICATION_JAVASCRIPT#, [11-65](#)
 - #BOX_BODY#, [11-64](#)
 - #DIALOG_CSS_CLASSES#, [11-66](#)
 - #DIALOG_HEIGHT#, [11-66](#)
 - #DIALOG_MAX_WIDTH#, [11-66](#)
 - #DIALOG_WIDTH#, [11-66](#)
 - #FAVICONS#, [11-66](#)
 - #FORM_CLOSE#, [11-65](#)
 - #FORM_OPEN#, [11-64](#)
 - #GLOBAL_NOTIFICATION#, [11-65](#)
 - #HEAD#, [11-66](#)
 - #IS_MODAL#, [11-66](#)
 - #LOGO#, [11-65](#)
 - #NAVIGATION_BAR#, [11-65](#)
- page template substitution strings (*continued*)
 - #NOTIFICATION_MESSAGE#, [11-64, 11-65](#)
 - #ONLOAD#, [11-66](#)
 - #PAGE_TRANSITION#, [11-66](#)
 - #REGION_POSITION_NN#, [11-64](#)
 - #SUCCESS_MESSAGE#, [11-65](#)
 - #TAB_CELLS#, [11-64, 11-65](#)
 - #TEMPLATE_JAVASCRIPT#, [11-65](#)
 - #THEME_STYLE_CSS#, [11-66](#)
 - #TITLE#, [11-64](#)
 - PAGE_URL, [11-66](#)
- page zero, [12-2](#)
- page-level item
 - about, [13-2](#)
 - All Page Items, [13-35](#)
 - Conditional Items, [13-35](#)
 - configuring Authorization Scheme, [13-29](#)
 - configuring Escape Special Characters, [13-17, 13-30](#)
 - configuring Restricted Characters, [13-17, 13-31](#)
 - configuring security, [13-16, 13-29](#)
 - configuring Session State Protection, [13-16, 13-30](#)
 - configuring Store value encrypted in session state, [13-17, 13-30](#)
 - creating, [13-1, 13-9, 13-22](#)
 - creating a shuttle, [13-6](#)
 - creating cascading list of values, [13-25](#)
 - creating cascading list of values (LOV), [13-13](#)
 - creating from Gallery, [13-10](#)
 - creating from Rendering tab, [13-10](#)
 - creating multiple using tabular form, [13-27](#)
 - creating static list of values, [13-23](#)
 - defining default values, [13-15, 13-29](#)
 - defining quick picks, [13-17, 13-31](#)
 - deleting multiple, [13-35](#)
 - displaying as conditional, [13-18, 13-32](#)
 - displaying as read only, [13-18, 13-32](#)
 - editing attributes, [13-28](#)
 - editing in Property Editor, [13-14](#)
 - editing multiple, [13-33](#)
 - Grid Edit of all Item Help Text, [13-35](#)
 - Grid Edit of all Item Labels, [13-35](#)
 - Item Help Subscriptions, [13-35](#)
 - multiple select list item, [13-4](#)
 - naming conventions, [13-2](#)
 - reassigning to a region, [13-34](#)
 - referencing using JavaScript, [13-4](#)
 - referencing values, [13-2](#)
 - specifying Authorization Scheme, [13-16](#)
 - static list of values, [13-11](#)
 - viewing, [13-9](#)
 - viewing History, [13-35](#)

- page-level item (*continued*)
 - viewing in Component view, [13-21](#)
 - viewing utilities, [13-20](#), [13-35](#)
 - Warn on Unsaved Changes, [13-19](#)
- page-level item reports
 - All Page Items, [13-20](#)
 - Conditional Items, [13-20](#)
 - Grid Edit of all Item Help Text, [13-20](#)
 - Grid Edit of all Item Labels, [13-20](#)
 - Item Help Subscriptions, [13-20](#)
- page-level template, [11-34](#)
- page, attaching authorization scheme, [19-64](#)
- pages
 - about, [2-2](#)
 - about Global page, [12-3](#)
 - calling from a button, [2-20](#)
 - calling with an alias, [2-19](#)
 - controlling access to, [19-44](#)
 - duplicate submission, [24-22](#)
 - enabling user customization, [12-8](#), [12-22](#)
 - exporting, [23-24](#)
 - linking with f?p syntax, [2-17](#)
 - optimizing for printing, [12-2](#)
 - resource use, [22-3](#)
 - searching for, [6-57](#)
 - server cache, [24-23](#)
 - specifying page template, [11-34](#)
- Pages Finder, [6-57](#)
- pagination
 - classic report, [8-104](#), [8-114](#)
 - interactive reports, [8-43](#), [8-68](#)
 - reports, [11-74](#), [11-79](#)
- Pagination Subtemplate
 - Next Page Template, [11-80](#)
 - Next Set Template, [11-80](#)
 - Pagination Template, [11-80](#)
 - Previous Page Template, [11-80](#)
 - Previous Set Template, [11-80](#)
- parent tabs, creating, [14-2](#)
- Parsing Schema, [6-31](#)
- password
 - best practices, [19-19](#)
 - changing, [1-23](#)
 - identifying risks, [19-19](#)
 - restricting reuse, [19-9](#)
 - Security Profiles Report, [19-20](#)
- passwords
 - Password Items Report, [19-20](#)
- patch, creating, [23-21](#)
- Percent Graph, [A-8](#)
- performance, [21-1](#)
 - best practices, [21-1](#)
 - database parameters, [21-6](#)
 - debugging problematic SQL queries, [21-3](#)
 - diagnosing issues, [21-2](#)
- performance (*continued*)
 - identifying issues, [21-2](#)
 - limiting resources, [21-6](#)
 - slow queries, [21-3](#)
 - slow running applications, [21-2](#)
 - uploading static files to Web server, [21-7](#)
 - using logs and report, [21-3](#)
- PL/SQL code, searching for, [6-59](#)
- PL/SQL Finder, [6-55](#)
- plug-ins
 - about, [18-4](#)
 - accessing, [18-5](#)
 - adding events, [18-10](#)
 - creating, [18-5](#)
 - custom attributes, [18-7](#)
 - deleting, [18-10](#)
 - editing, [18-7](#)
 - examples, [18-4](#)
 - exporting, [18-12](#)
 - implementing, [18-3](#)
 - importing, [18-11](#), [23-33](#)
 - resetting, [18-12](#)
 - uploading associated files, [18-8](#)
 - using JavaScript code, [18-18](#)
 - utilization, [18-12](#)
 - viewing history, [18-13](#)
 - viewing Plug-in Repository, [18-11](#)
- Popup LOV, [A-9](#)
- Popup LOV template
 - Buttons, [11-69](#)
 - editing, [11-67](#)
 - Page Attributes, [11-69](#)
 - Pagination, [11-69](#)
 - Result Set, [11-69](#)
 - Search Field, [11-68](#)
 - Window, [11-69](#)
- popup LOV, creating dependent LOV, [13-25](#)
- preconfigured authentication schemes, [19-48](#)
- Printer Friendly template, [12-2](#)
- printing reports, [8-123](#)
- privileges
 - about granting to database users, [D-1](#)
 - granted to database users, [D-1](#)
- Procedure Finder, [6-59](#)
- procedures, searching for, [6-59](#)
- process
 - Cache Management, [15-23](#), [15-27](#), [15-28](#)
 - Data Manipulation, [15-23](#), [15-27](#), [15-28](#)
 - Form Pagination, [15-23](#), [15-27](#), [15-28](#)
 - On Demand, [15-23](#), [15-27](#), [15-28](#)
 - PL/SQL, [15-23](#), [15-27](#), [15-28](#)
 - Report Pagination, [15-23](#), [15-27](#), [15-28](#)
 - Web Services, [15-23](#), [15-27](#), [15-28](#)
- productivity application, unlocking, [3-7](#)
- programming techniques

programming techniques (*continued*)
 collections, [17-7](#)
 database links, [17-30](#)
 implementing Web services, [17-8](#)
 proxy server, [17-10](#)
 Proxy Server attribute, [6-19](#)
 public file upload, enabling, [19-5](#)
 public synonyms, [D-1](#)
 Public User attribute, [6-25](#)

Q

Query Finder, [6-55](#), [6-58](#)
 query, searching for, [6-58](#)

R

radio group, [A-10](#)
 LOV, [13-25](#)
 Recently Updated Pages report, [6-77](#)
 reflow reports
 creating with Create Application Wizard, [6-4](#),
[12-16](#)
 region
 about types, [12-13](#)
 accessing utilities and reports, [12-26](#)
 adding an image, [12-11](#), [12-25](#)
 adding sliders to regions, [16-11](#)
 based on a URL, [12-30](#)
 controlling positioning, [12-8](#), [12-21](#)
 copying, [12-12](#), [12-28](#)
 creating, [12-5](#), [12-18](#)
 creating for a list, [16-39](#)
 creating region display selector, [12-11](#), [12-28](#)
 creating subregion, [12-26](#)
 deleting, [12-13](#), [12-29](#)
 deleting multiple regions at once, [12-29](#)
 displaying conditionally, [12-21](#)
 editing attributes, [12-7](#), [12-19](#), [12-27](#)
 editing attributes for multiple regions, [12-25](#)
 editing region attributes, [12-7](#), [12-19](#), [12-27](#)
 enabling customization, [12-8](#), [12-22](#)
 enabling users to customize a page, [12-8](#),
[12-22](#)
 how attributes affect layout, [12-19](#)
 specifying a static region ID, [12-10](#), [12-24](#)
 specifying header and footer, [12-8](#), [12-21](#)
 understanding, [12-5](#), [12-13](#)
 region attributes
 Cache, [12-9](#), [12-22](#)
 Column, [12-20](#)
 Column Attributes, [12-20](#)
 Column CSS Classes, [12-20](#)
 Column Span, [12-20](#)
 Condition Type, [12-21](#)

region attributes (*continued*)
 Customization, [12-21](#)
 Display Point, [12-20](#)
 Expression, [12-21](#)
 Image Tag Attributes, [12-20](#)
 Item Display Position, [12-20](#)
 Read Only Condition Type, [12-21](#)
 Region CSS Classes, [12-20](#)
 Region Display Selector, [12-20](#)
 Region Footer, [12-21](#)
 Region Header, [12-21](#)
 Region Image, [12-20](#)
 Sequence, [12-19](#)
 Start New Grid, [12-20](#)
 Static ID, [12-10](#), [12-24](#)
 Template, [12-20](#)
 region display selector, creating, [12-11](#), [12-28](#)
 region template
 Definition, [11-70](#)
 editing, [11-69](#)
 Form Table Attributes, [11-71](#)
 Name, [11-70](#)
 Sub Regions Attributes, [11-71](#)
 Substitution Strings, [11-71](#)
 region titles
 marking as translatable, [20-4](#)
 region type
 Breadcrumb, [12-15](#), [12-17](#)
 Calendar, [12-15](#), [12-17](#)
 Chart, [12-15](#), [12-17](#)
 Classic Report, [12-14](#), [12-16](#)
 Form and Report on Web Service, [12-15](#),
[12-17](#)
 Form on a Procedure, [12-14](#), [12-16](#)
 Form on a SQL Query, [12-14](#), [12-17](#)
 Form on a Table, [12-14](#)
 Form on a Table with List View, [12-17](#)
 Form on Web Service, [12-17](#)
 Help Text, [12-15](#), [12-18](#)
 Interactive Report, [12-14](#)
 List, [12-15](#)
 List View, [12-16](#)
 Map, [12-15](#)
 Multiple Static Content, [12-14](#), [12-16](#)
 PL/SQL Dynamic Content, [12-15](#), [12-17](#)
 Plug-ins, [12-15](#), [12-17](#)
 Region Display Selector, [12-15](#), [12-18](#)
 Report on Web Service Result, [12-14](#), [12-16](#)
 Static Content, [12-14](#), [12-16](#)
 Summary Page, [12-17](#)
 Tree, [12-15](#)
 URL, [12-15](#), [12-18](#)
 Rejoin Sessions
 about security risks, [19-7](#)
 configuring at application-level, [6-27](#)

- Rejoin Sessions (*continued*)
 - configuring at page-level, [7-34](#), [24-26](#)
- Rejoin Sessions attribute, [6-27](#)
- report, [8-6](#)
 - creating with Create Application Wizard, [6-3](#)
 - Oracle database, [21-6](#)
- report and form
 - creating with Create Application Wizard, [6-3](#)
- report column templates
 - colorizing cell headings, [11-74](#)
 - omitting headings, [11-74](#)
- report column templates (generic)
 - After Rows, [11-76](#)
 - Before Each Row, [11-74](#), [11-75](#)
 - Before Rows, [11-74](#)
 - Column Headings, [11-74](#)
 - Column Templates, [11-75](#)
 - displaying HTML after all columns, [11-75](#)
 - displaying HTML after last row, [11-76](#)
 - Next Page Template, [11-77](#)
 - Next Set Template, [11-77](#)
 - Pagination Subtemplate, [11-76](#)
 - Previous Page Template, [11-77](#)
 - Previous Set Template, [11-77](#)
 - Report Template, [11-73](#)
 - Row Highlighting, [11-76](#)
- report column templates (named)
 - Before Rows attribute, [11-79](#)
 - colorizing column header cell, [11-79](#)
 - Column Headings, [11-79](#)
 - Pagination Subtemplate, [11-79](#)
 - Report Template Identification, [11-77](#)
 - Row Templates, [11-78](#)
 - Subscription, [11-78](#)
- report layouts
 - about, [16-49](#)
 - copying, [16-52](#)
 - creating, [16-51](#)
 - editing, [16-52](#)
 - options, [16-50](#)
- report on a form on table, [8-4](#)
- Report on Web service result
 - creating with Create Region Wizard, [12-14](#), [12-16](#)
- report queries
 - about, [16-46](#)
 - copying, [16-49](#)
 - creating, [16-46](#)
 - editing, [16-49](#)
- report regions
 - about configuration options, [8-124](#)
 - printing, [8-123](#)
 - printing to PDF, [8-124](#)
- report templates, [11-72](#)
 - editing, [11-72](#)
- report templates (*continued*)
 - generic column templates, [11-72](#)
 - named column templates, [11-73](#)
 - using JavaScript, [11-80](#)
- reports, [11-42](#)
 - about, [8-1](#)
 - about database application, [6-74](#)
 - Active Session History (ASH) reports, [21-6](#)
 - adding by dragging and dropping, [8-11](#)
 - Application Dashboard, [6-75](#), [21-4](#)
 - Automatic Workload Repository (AWR) reports, [21-6](#)
 - Bulk Edit Item Help, [10-38](#)
 - Change History, [6-77](#)
 - class references, [11-30](#)
 - creating a column link, [8-107](#), [8-119](#)
 - creating using wizard, [8-5](#)
 - creating with Create Application Wizard, [8-9](#)
 - creating with Create Page Wizard, [8-11](#)
 - CSV export, [11-74](#)
 - database applications, [6-74](#)
 - exporting to CSV format, [11-74](#), [11-79](#)
 - including pagination above, [11-74](#), [11-79](#)
 - managing interactive grids, [8-13](#)
 - monitor activity, [21-5](#)
 - performance, [21-3](#)
 - printing, [8-123](#)
 - Recently Updated Pages, [6-77](#)
 - report and form, [8-4](#)
 - sorting columns, [8-107](#), [8-117](#)
 - supported substitution strings, [11-31](#)
 - template subscription, [11-42](#)
 - template utilization, [11-42](#)
 - templates in a theme, [11-29](#)
 - theme file references, [11-30](#)
 - theme template counts, [11-29](#)
 - themes, [11-29](#)
 - understanding types, [8-1](#)
 - workspace activity, [21-4](#), [21-5](#)
- REQUEST
 - about scope and value, [2-44](#)
 - about When Button Pressed attribute, [2-44](#)
 - button names, [13-61](#)
 - referencing using declarative conditions, [2-44](#)
 - referencing value of, [2-43](#)
 - substitution string, [2-43](#)
 - using for Show processing, [2-44](#)
- resource use, monitoring, [22-3](#)
- responsive design, [2-6](#)
- responsive design, about, [11-2](#)
- RESTful access, enabling, [19-7](#)
- restrict access, to application, [6-21](#)
- Run Application button, [1-18](#)
- Run on Public Pages attribute, [6-26](#)
- Runtime API Usage, [6-31](#)

Runtime Developer toolbar, [7-38](#), [24-29](#)
 Application, [7-38](#), [24-29](#)
 Debug, [7-38](#), [24-29](#)
 Developer Toolbar Options, [7-38](#), [24-29](#)
 Edit Page, [7-38](#), [24-29](#)
 Home, [7-38](#), [24-29](#)
 Quick Edit, [7-38](#), [11-40](#), [24-29](#)
 Session, [7-38](#), [24-29](#)
 Show Layout Columns, [7-38](#), [24-29](#)

runtime environment, [1-3](#)
 advantages, [19-14](#)

S

safe item types, [19-22](#)

sample charts
 viewing sample, [10-24](#)

Sample Database Application
 about, [3-8](#)
 Administration page, [3-15](#)
 Customers page, [3-11](#)
 editing, [3-17](#), [3-18](#)
 editing an application, [3-18](#)
 Home page, [3-10](#)
 installing, [3-8](#)
 Mobile page, [3-16](#)
 modifying, [3-17](#), [3-18](#)
 running, [3-8](#)
 understanding, [3-8](#), [3-9](#)

Save Public Report Authorization, [8-25](#)

SDLC
 rapid application development, [23-4](#)
 Spiral, [23-4](#)

Search Workspace, [1-7](#)

secure sockets layer, best practices, [19-6](#)

security
 attributes, [6-23](#)
 for pages, [24-19](#)

security attributes
 Allows URLs Created After, [6-29](#)
 authentication, [6-24](#)
 Authentication Scheme, [6-25](#)
 Authorization Scheme, [6-25](#)
 Bookmark Hash Function, [6-29](#)
 Browser Security, [6-30](#)
 Cache, [6-30](#)
 Database Session, [6-31](#)
 Deep Linking, [6-27](#)
 Define Authorization Schemes, [6-25](#)
 Embed in Frames, [6-30](#)
 Expire Bookmarks, [6-29](#)
 HTML Escaping Mode, [6-30](#)
 HTTP Response Headers, [6-31](#)
 Manage Session State Protection, [6-29](#)

security attributes (*continued*)
 Maximum Session Idle Time in Seconds, [6-28](#)
 Maximum Session Length in Seconds, [6-28](#)
 Public User, [6-25](#)
 Rejoin Sessions, [6-27](#)
 Run on Public Pages, [6-26](#)
 Session Idle Timeout URL, [6-28](#)
 Session Management, [6-26](#)
 Session State Protection, [6-29](#)
 Session Timeout URL, [6-28](#)

security best practices
 configuring instance settings, [19-10](#)
 configuring storage options, [19-12](#)
 configuring workspace settings, [19-12](#)
 controlling SQL access in Websheets, [19-10](#)
 creating login controls, [19-5](#)
 creating wallets, [19-12](#)
 cross site-scripting protection, [19-21](#)
 embedded PL/SQL gateway, [19-4](#)
 enabling email, [19-11](#)
 enabling public file upload, [19-5](#)
 enabling RESTful access, [19-7](#)
 enabling RESTful services, [19-10](#)
 enabling self service workspace provisioning, [19-11](#)
 for administrators, [19-1](#)
 for developers, [19-18](#)
 identifying risk passwords, [19-19](#)
 isolating workspaces, [19-8](#)
 items of type password, [19-19](#)
 managing component availability, [19-13](#)
 managing instance security, [19-4](#)
 managing users, [19-13](#)
 managing workspace to schema assignments, [19-13](#)
 managing workspaces, [19-12](#)
 Oracle BI Publisher, [19-14](#)
 Oracle HTTP Server, [19-3](#)
 restricting access by IP address, [19-5](#)
 restricting password reuse, [19-9](#)
 runtime environment, [19-14](#)
 secure sockets layer, [19-6](#)
 session joins, [19-7](#)
 session state, [19-26](#)
 session timeout, [19-8](#)
 specifying instance proxy, [19-6](#)

security risks, file upload table, [19-39](#)

seeding, [20-9](#)

select list, [A-12](#)

select lists, creating dependent LOV, [13-25](#)

session cloning
 enabling, [2-25](#)

session ID, [2-14](#)

Session Idle Timeout URL, [6-28](#)

- session joins
 - preventing browser attacks, [19-7](#)
 - security best practices, [19-7](#)
- Session Management, [6-26](#)
 - specifying, [6-26](#)
- session state
 - clearing, [2-22](#)
 - clearing application cache, [2-24](#)
 - clearing cache by item, [2-22](#)
 - clearing cache by page, [2-22](#)
 - clearing cache for current user session, [2-24](#)
 - clearing cache for two pages, [2-23](#)
 - enabling projection, [6-29](#)
 - management, [2-13](#)
 - passing item value, [2-23](#)
 - protection, [19-26](#)
 - referencing, [2-20](#)
 - resetting an application completely, [2-24](#)
 - security best practices, [19-26](#)
 - setting, [2-21](#)
 - sharing across applications, [16-23](#)
 - viewing, [2-14](#)
- session state protection
 - configuring all attributes, [19-30](#)
 - configuring security attributes, [19-29](#)
 - disabling, [19-27](#)
 - enabling, [19-27](#)
 - enabling from Edit Security Attributes, [19-27](#)
 - enabling from Session State Protection, [19-28](#)
 - enabling using a wizard, [19-28](#)
 - reports, [19-29](#)
 - understanding, [19-27](#)
 - viewing existing settings, [19-29](#)
 - viewing summaries, [19-29](#)
- Session State Protection attribute, [6-29](#)
- session state values, managing, [2-20](#)
- Session Timeout URL, [6-28](#)
- session timeout, utilizing, [19-8](#)
- shared components
 - about, [16-3](#)
 - accessing, [16-2](#)
 - backing up, [23-21](#)
 - creating, [16-2](#)
 - exporting, [23-21](#)
 - Shared Components page, [16-1](#)
- Shared Components icon, [1-18](#)
- Shared Components page
 - Application Computations, [16-4](#)
 - Application Definition Attributes, [16-3](#)
 - Application Items, [16-3](#)
 - Application Processes, [15-32](#), [16-4](#)
 - Authentication Schemes, [16-4](#)
 - Authorization Schemes, [16-4](#)
 - Breadcrumbs, [16-6](#)
- Shared Components page (*continued*)
 - Build Options, [16-5](#)
 - Component Settings, [16-5](#)
 - Data Load Definitions, [16-7](#)
 - Developer Comments Calendar, [16-8](#)
 - Edit Application Comments, [16-8](#)
 - Export Application Components, [16-8](#)
 - Globalization Attributes, [16-8](#)
 - List of Values, [16-5](#)
 - Lists, [16-6](#)
 - Manage Supporting Objects, [16-8](#)
 - Navigation Bar Entries, [16-6](#)
 - Navigation Menu, [16-6](#)
 - Plug-ins, [16-5](#)
 - Report Layouts, [16-8](#)
 - Report Queries, [16-8](#)
 - Review Developer Comments, [16-8](#)
 - Security Attributes, [16-4](#)
 - Session State Protection, [16-4](#)
 - Shortcuts, [16-5](#)
 - Static Application Files, [16-7](#)
 - Static Workspace Files, [16-7](#)
 - Tabs, [16-6](#)
 - Tasks Region, [16-8](#)
 - Templates, [16-6](#)
 - Text Messages, [16-8](#)
 - Themes, [16-6](#)
 - Translate Application, [16-8](#)
 - User Interface Attributes, [16-6](#)
 - Web Service References, [16-7](#)
- shared interactive reports, linking to, [8-91](#)
- shortcuts, [16-42](#)
 - about, [16-42](#)
 - copying an existing shortcut, [16-44](#)
 - defining, [16-43](#), [16-44](#)
 - defining from scratch, [16-43](#)
 - editing attributes, [16-45](#)
 - History report, [16-45](#)
 - HTML Text, [16-42](#), [16-44](#)
 - HTML Text with Escaped Special Characters, [16-42](#), [16-44](#)
 - Image, [16-42](#), [16-44](#)
 - Message, [16-43](#), [16-44](#)
 - Message with JavaScript Escaped Special Quotes, [16-43](#), [16-44](#)
 - PL/SQL Function Body, [16-42](#), [16-44](#)
 - referencing, [16-45](#)
 - reports, [16-45](#)
 - Subscription report, [16-45](#)
 - supporting translatable messages, [20-3](#)
 - Text with JavaScript Escaped Single Quotes, [16-42](#), [16-44](#)
 - types, [16-42](#)
- Show Page, [2-10](#)
- shuttle multiple select list, [A-13](#)

- shuttle, creating dependent LOV, [13-25](#)
- sign in page, [1-1](#)
- Single Sign-On
 - setting up, [19-57](#)
- Slider item type, [A-14](#)
- SOAP, [17-8](#)
- SQL tracing, enabling, [22-3](#)
- standard tabs, creating, [14-2](#)
- static application files
 - deleting, [16-16](#)
 - deleting all, [16-16](#)
 - downloading, [16-15](#)
 - downloading in Zip file, [16-15](#)
 - managing, [16-12](#)
 - referencing using #APP_IMAGES#, [16-14](#)
 - uploading, [16-13](#)
 - viewing details, [16-14](#)
- static files
 - exporting, [23-25](#)
 - importing into another instance, [23-25](#)
- static list
 - about, [16-29](#)
 - adding list entries, [16-31](#)
 - adding sublists, [16-31](#)
 - copying entries, [16-33](#)
 - creating from scratch, [16-30](#)
 - managing orphaned list entries, [16-34](#)
 - reparenting entries, [16-34](#)
 - resequencing entries, [16-35](#)
- static substitution string, [6-23](#)
- static workspace files
 - deleting, [16-19](#)
 - deleting all, [16-20](#)
 - downloading, [16-19](#)
 - downloading all in Zip file, [16-19](#)
 - managing, [16-17](#)
 - referencing using
 - #WORKSPACE_IMAGES#, [16-18](#)
 - uploading, [16-17](#)
 - viewing details, [16-18](#)
- Status attribute, [6-20](#)
- storage options, configuring, [19-12](#)
- style sheet, [11-32](#)
- subregion, [12-26](#)
 - creating, [12-27](#)
 - creating by editing attributes, [12-27](#)
 - using, [12-26](#)
- substitution strings, [2-31](#)
 - #CSV_LINK#, [11-74](#), [11-79](#)
 - #GLOBAL_NOTIFICATION#, [6-22](#)
 - about built-in, [2-28](#)
 - APEX\$ROW_NUM, [2-29](#)
 - APEX\$ROW_SELECTOR, [2-30](#)
 - APEX\$ROW_STATUS, [2-30](#)
 - APP_ALIAS, [2-31](#)
- substitution strings (*continued*)
 - APP_DATE_TIME_FORMAT, [2-32](#)
 - APP_ID, [2-30](#)
 - APP_IMAGES, [2-32](#)
 - APP-NLS_DATE_FORMAT, [2-33](#)
 - APP-NLS_TIMESTAMP_FORMAT, [2-33](#)
 - APP-NLS_TIMESTAMP_TZ_FORMAT, [2-34](#)
 - APP_PAGE_ALIAS, [2-34](#)
 - APP_PAGE_ID, [2-34](#)
 - APP_REQUEST_DATA_HASH, [2-35](#)
 - APP_SESSION, [2-35](#)
 - APP_SESSION_VISIBLE, [2-36](#)
 - APP_TITLE, [2-36](#)
 - APP_UNIQUE_PAGE_ID, [2-37](#)
 - APP_USER, [2-37](#)
 - Application Express SCHEMA OWNER, [2-45](#)
 - AUTHENTICATED_URL_PREFIX, [2-38](#)
 - BROWSER_LANGUAGE, [2-39](#)
 - calendar, [24-67](#)
 - CURRENT_PARENT_TAB_TEXT, [2-39](#)
 - DEBUG, [2-39](#)
 - for application items, [2-26](#)
 - for page items, [2-26](#)
 - IMAGE_PREFIX, [2-40](#)
 - in page templates, [11-59](#)
 - JET_BASE_DIRECTORY, [2-41](#)
 - JET_CSS_DIRECTORY, [2-41](#)
 - JET_JS_DIRECTORY, [2-41](#)
 - LOGOUT_URL, [2-42](#)
 - output escaping, [2-27](#), [11-33](#)
 - PRINTER_FRIENDLY, [2-42](#)
 - PROXY_SERVER, [2-42](#)
 - PUBLIC_URL_PREFIX, [2-43](#)
 - report of supported, [11-31](#)
 - REQUEST, [2-43](#)
 - SQLERRM, [2-45](#)
 - static, [6-23](#)
 - supported in region footer, [12-8](#), [12-21](#)
 - SYSDATE_YYYYMMDD, [2-45](#)
 - THEME_DB_IMAGES, [2-46](#)
 - THEME_IMAGES, [2-46](#)
 - understanding, [2-26](#)
 - usage within templates, [2-28](#)
 - using, [2-26](#)
 - within templates, [2-26](#)
 - WORKSPACE_ID, [2-47](#)
 - WORKSPACE_IMAGES, [2-46](#)
- success message, creating for a page process, [15-25](#), [15-30](#)
- Supported Charts
 - Oracle JavaScript Extension Toolkit (Oracle JET), [10-24](#)
- supporting objects
 - creating, [23-9](#)
 - deinstalling, [23-15](#)

Supporting Objects icon, [1-18](#)
 SVG chart
 displaying in other languages, [24-50](#)
 migrating to Flash, [24-46](#)
 referencing CSS styles inline, [24-50](#)
 referencing custom CSS, [24-49](#)
 upgrade restrictions, [24-45](#)
 upgrading all to HTML5, [24-46](#)
 upgrading to HTML5, [24-46](#)
 SVG chart, about CSS classes, [24-46](#)
 Switch
 configuring, [16-11](#)
 Switch item type, [A-13](#)
 syntax, for dynamic list, [16-36](#)
 system message, [1-5](#)

T

tab

 about Tabs page, [14-3](#)
 adding, [14-2](#)
 Conditional Display report, [14-8](#)
 create Standard tab, [14-7](#)
 create Standard tab set, [14-7](#)
 creating, [14-3](#), [14-5](#)
 editing, [14-7](#)
 editing multiple, [14-7](#)
 History report, [14-8](#)
 managing, [14-3](#)
 move standard to different parent, [14-7](#)
 Parent, [14-3](#)
 rename standard, [14-7](#)
 reports, [14-8](#)
 resequence, [14-7](#)
 Standard, [14-2](#)
 Standard Tab Task, [14-7](#)
 template support, [14-4](#)
 Utilization report, [14-8](#)

tab reports

 Conditional Display, [14-8](#)
 History, [14-8](#)

Table Finder, [6-55](#)

Tables Finder, [6-59](#)

tables, searching for, [6-59](#)

Tabs page, [14-3](#)

tabular form

 creating, [24-56](#)
 creating a page process, [15-28](#)

Tag Cloud, [4-3](#)

Team Development

 accessing, [4-1](#)
 Features dashboard, [4-4](#), [4-7](#)
 home page, [4-2](#)
 managing bugs, [4-16](#)
 managing feedback, [4-20](#), [4-23](#)

Team Development (*continued*)

 managing links, [4-29](#)
 managing news entries, [4-28](#)
 managing To Dos, [4-13](#)
 News Region, [4-2](#)
 purging data, [4-28](#)
 tracking features, [4-3](#)
 tracking milestones, [4-10](#)
 Utilities, [4-2](#)

Team Development file attachments

 viewing, [4-27](#)

Team Development utilities, [4-24](#)

 accessing, [4-24](#)
 assigning milestones, [4-27](#)
 change milestones for overdue features, [4-27](#)
 configuring Team Development settings, [4-26](#)

 editing focus areas, [4-27](#)

 enable support for file upload, [4-24](#)

 manage focus areas, [4-24](#)

 manage links, [4-24](#)

 manage news, [4-24](#)

 manage uploaded files, [4-24](#)

 managing links, [4-29](#)

 managing news entries, [4-28](#)

 Purge Data, [4-24](#)

 purging data, [4-28](#)

 push due date of open features, [4-27](#)

 push past due bugs, [4-24](#)

 Release Summary, [4-26](#)

 Release Summary report, [4-24](#)

 set feature due date to milestone, [4-27](#)

 Team Development settings, [4-24](#)

 update assignee, [4-24](#)

 viewing file attachments, [4-27](#)

template edit history, [11-42](#)

template option presets, [11-9](#)

template options

 about, [11-8](#)

 default, [11-9](#)

 groups, [11-9](#)

 presets, [11-9](#)

 using, [11-8](#)

template options groups, [11-9](#)

templates

 breadcrumbs, [11-47](#)

 button, [11-49](#)

 changing defaults, [11-10](#)

 columns, [11-75](#), [11-80](#)

 creating, [11-35](#)

 customizing, [11-32](#)

 labels, [11-53](#)

 legacy calendar, [11-51](#)

 lists, [11-56](#)

 managing, [11-43](#)

- templates (*continued*)
 - page, [11-59](#)
 - popup LOV, [11-67](#)
 - publishing, [11-46](#)
 - regions, [11-69](#)
 - replacing, [11-44](#)
 - replacing all, [11-45](#)
 - reports, [11-72](#)
 - selecting for a page, [11-34](#)
 - selecting theme default, [11-34](#)
 - supported substitution strings, [11-31](#)
 - unsubscribing to, [11-46](#)
 - viewing for specific page, [11-44](#)
 - viewing History report, [11-42](#)
 - viewing on Templates page, [11-43](#)
 - viewing Publish report, [11-43](#)
 - viewing region position utilization, [11-45](#)
 - viewing reports, [11-42](#)
 - viewing Subscription report, [11-42](#)
 - viewing Utilization report, [11-42](#)
- text area item type, [A-17](#)
- Text Field item type, [A-15](#)
- text strings, [20-35](#)
 - translating, [20-16](#)
- text strings, translating, [20-15](#)
- Text Text Field with autocomplete item type, [A-16](#)
- theme attributes, [11-11](#)
 - Breadcrumb, [11-13](#), [11-14](#)
 - Button, [11-13](#)
 - Charts, [11-14](#)
 - Classic Report, [11-14](#)
 - CSS File URLs, [11-12](#)
 - Custom Classes, [11-16](#)
 - Custom Image, [11-16](#)
 - Custom Library File URLs, [11-16](#)
 - Custom Prefix Class, [11-16](#)
 - Date Picker Icon Attributes, [11-16](#)
 - Date Picker Icon Name, [11-16](#)
 - Default Label, [11-13](#)
 - Dialog Button Region, [11-14](#)
 - Dialog Content Region, [11-14](#)
 - Dialog Page, [11-15](#)
 - Error Page, [11-13](#)
 - Files, [11-16](#)
 - Footer Toolbar, [11-14](#)
 - Forms, [11-14](#)
 - Global Template Options, [11-15](#)
 - Header Toolbar, [11-14](#)
 - Interactive Reports, [11-14](#)
 - JavaScript File URLs, [11-12](#)
 - Legacy Calendar, [11-13](#)
 - Library, [11-15](#)
 - List, [11-13](#)
 - Lists, [11-14](#)
- theme attributes (*continued*)
 - Login Page, [11-13](#)
 - Name, Application, [11-11](#)
 - Name, Description, [11-11](#)
 - Name, Name, [11-11](#)
 - Name, Navigation Bar Implementation, [11-11](#)
 - Name, Navigation Type, [11-11](#)
 - Name, Theme Number, [11-11](#)
 - Name, User Interface, [11-11](#)
 - Navigation Bar List, [11-13](#)
 - Navigation Menu List (Top), [11-13](#)
 - Navigation Menu List Position, [11-13](#)
 - Optional Label, [11-13](#)
 - Page, [11-12](#)
 - Page Template Options, [11-13](#)
 - Page Transition, [11-15](#)
 - Popup Transition, [11-15](#)
 - Printer Friendly Page, [11-13](#)
 - Region, [11-13](#)
 - Report, [11-14](#)
 - Required Label, [11-13](#)
 - Styles, [11-16](#)
 - Tabular Forms, [11-14](#)
 - Theme Subscription, [11-12](#)
 - Wizards, [11-14](#)
- theme reports
 - Application Templates, [11-29](#)
 - class references, [11-30](#)
 - Class References, [11-30](#)
 - File References, [11-30](#)
 - files references, [11-30](#)
 - supported substitution strings, [11-31](#)
 - template counts, [11-29](#)
 - Template Substitution Strings, [11-31](#)
- templates in a theme, [11-29](#)
- Theme Template Counts, [11-29](#)
- Theme Roller, [7-38](#), [24-29](#)
 - using, [11-6](#)
- theme styles, [11-3](#)
 - about, [11-3](#)
 - creating, [11-4](#)
 - deleting, [11-5](#)
 - editing, [11-5](#)
 - editing from User Interface page, [11-5](#)
 - Theme Roller, [11-6](#)
 - using, [11-1](#)
- theme subscriptions
 - about, [11-17](#), [11-18](#)
 - modifying, [11-18](#)
 - re-subscribing, [11-20](#)
 - refreshing, [11-19](#)
 - unsubscribing, [11-19](#)
 - verifying refresh, [11-18](#)
- themes
 - about, [11-1](#)

- themes (*continued*)
 - about subscriptions, [11-17](#)
 - accessing, [11-9](#)
 - changing ID, [11-23](#)
 - changing identification number (ID), [11-23](#)
 - copying, [11-24](#)
 - creating, [11-20](#)
 - creating master application, [11-25](#)
 - default templates, [11-10](#)
 - deleting, [11-24](#)
 - editing, [11-10](#)
 - editing theme attributes, [11-11](#)
 - exporting, [11-28](#), [23-26](#)
 - exporting and importing, [11-28](#)
 - exporting with different ID, [11-23](#)
 - grid-based, [11-3](#)
 - importing, [11-28](#), [23-33](#)
 - live template options, [11-40](#)
 - managing, [11-1](#)
 - reports, [11-29](#)
 - selecting page-level template, [11-34](#)
 - switching, [11-21](#)
 - switching to Universal Theme, [11-22](#)
 - theme styles, [11-3](#)
 - Themes page, [11-9](#)
 - using, [11-1](#)
 - Themes page
 - accessing from Shared Components, [11-9](#)
 - To Do reports
 - Calendar page, [4-16](#)
 - Progress Log, [4-16](#)
 - To Dos
 - attaching files, [4-14](#)
 - creating, [4-13](#)
 - managing, [4-13](#)
 - To Dos page, [4-13](#)
 - updating a task, [4-14](#)
 - viewing, [4-13](#)
 - viewing reports, [4-15](#)
 - translatable messages, defining, [20-16](#)
 - translating, [20-35](#)
 - messages in PL/SQL procedures, [20-16](#)
 - translation, [20-1](#)
 - applications, [20-1](#)
 - dynamic, [20-58](#)
 - dynamic text strings, [20-3](#)
 - dynamic translations, [20-4](#)
 - editing manually, [20-14](#)
 - exporting text, [20-9](#)
 - globalization support, [20-1](#)
 - language identification, [20-2](#)
 - mapping primary application ID, [20-8](#)
 - mapping target application ID, [20-8](#)
 - messages, [20-3](#)
 - region titles, [20-4](#)
 - translation (*continued*)
 - rules, [20-2](#)
 - seeding, [20-9](#)
 - shortcuts, [20-3](#)
 - steps, [20-7](#)
 - templates, [20-5](#)
 - translation file, [20-9](#)
 - understanding, [20-7](#)
 - understanding application rendering, [20-2](#)
 - XLIFF, [20-9](#)
 - XLIFF Target Elements, [20-11](#)
 - translation file, [20-9](#)
 - publishing, [20-12](#)
 - uploading, [20-12](#)
 - trees
 - about, [13-62](#)
 - creating, [13-61](#), [13-65](#), [13-69](#)
 - creating on new page, [13-63](#)
 - editing attributes, [13-67](#), [13-70](#)
 - managing in Component View, [13-68](#)
 - managing in Page Designer, [13-62](#)
- ## U
-
- Universal Theme
 - Migration Guide, [11-22](#)
 - switching to, [11-22](#)
 - Universal Theme - 42, [11-2](#), [11-3](#)
 - unlocking, productivity application, [3-7](#)
 - unsaved changes
 - warning users, [7-36](#), [13-19](#), [13-54](#)
 - Upgrade Application icon, [6-76](#), [6-77](#)
 - URL
 - publishing, [23-38](#), [23-39](#)
 - SSL-enabled, [12-30](#)
 - syntax, [2-16](#)
 - user identity
 - establishing, [19-45](#)
 - verifying, [2-12](#)
 - user interface
 - about, [1-2](#)
 - about mobile applications, [2-6](#)
 - about selecting, [2-5](#)
 - controlling, [11-1](#)
 - managing design of, [2-5](#)
 - User Interface attributes
 - Add New User Interface, [6-39](#)
 - configuring, [6-36](#)
 - CSS file URLs, [6-39](#)
 - Favicon, [6-39](#)
 - Favicon HTML, [6-39](#)
 - Image Prefix, [6-38](#)
 - Logo, [6-38](#)
 - Media Type, [6-38](#)
 - Static File Prefix, [6-38](#)

User Interface attributes (*continued*)

- User Interface Detection, [6-39](#)
- User Interfaces, [6-39](#)

User Interface Defaults

- exporting, [23-27](#)
- importing, [23-34](#)

User Interface Details, [6-40](#)

- Auto Detect, [6-41](#)
- Concatenated Files, [6-44](#)
- Content Delivery Network, [6-43](#)
- Default, [6-41](#)
- Display Name, [6-41](#)
- Enable End Users to choose Theme Style, [6-41](#)
- File URLs, [6-43](#), [6-44](#)
- Global Page, [6-42](#)
- Home URL, [6-41](#)
- Include jQuery Migrate, [6-44](#)
- Include Legacy JavaScript, [6-44](#)
- Implementation, [6-42](#)
- List Template, [6-42](#), [6-43](#)
- Login URL, [6-41](#)
- Navigation Bar List, [6-43](#)
- Navigation Menu List, [6-42](#)
- Position, [6-42](#)
- Sequence, [6-41](#)
- Template Options, [6-42](#), [6-43](#)
- Theme, [6-41](#)
- Theme Style, [6-42](#)
- User Interface Type, [6-41](#)

user preferences

- default schema, [1-23](#)
- editing, [1-23](#)
- enabling legacy Component View, [1-23](#)
- running in new window, [1-23](#)
- using single window, [1-23](#)

user roles

- developer, [19-2](#)
- Oracle Application Express administrator, [19-2](#)
- Workspace administrator, [19-2](#)

users

- about managing, [19-13](#)

utilities

- accessing for application, [6-74](#)

Utilities icon, [1-18](#)

V

V function, [2-21](#)

validations

- about, [15-8](#)
- associating with item, [9-22](#), [9-23](#)
- creating, [15-9](#), [15-10](#)
- creating for item, [15-8](#)

validations (*continued*)

- creating for page, [15-8](#), [15-14](#)
- creating for page item, [15-11](#)
- creating for tabular form column, [15-16](#)
- creating for tabular form row, [15-17](#)
- creating with error messages, [9-22](#)
- defining error messages, [15-20](#)
- editing, [15-18](#)
- Go to Error link, [15-21](#)
- making conditional, [15-20](#)
- When Button Pressed attribute, [2-44](#)

Version attribute, [6-18](#)

view, viewing data within, [6-78](#)

W

wallets, creating, [19-12](#)

Warn on Unsaved Changes, [7-36](#), [13-19](#)

Web service process

- editing a process, [17-26](#)
- mapping input parameters to static values, [17-26](#)

Web service reference, [17-9](#)

- creating, [17-10](#)
- creating forms, [17-20](#)
- creating forms and reports, [17-17](#)
- creating manually, [17-12](#), [17-14](#)
- specifying WSDL, [17-11](#)
- viewing a history, [17-27](#)

Web Service Reference page, [17-9](#)

Web services, [17-8](#)

- creating a reference from a WSDL, [17-12](#)
- creating RESTful Web services, [17-15](#)
- displaying Web service results in a report, [17-25](#)
- editing a Web service process, [17-26](#)
- invoking as a process, [17-23](#)
- RESTful Web service
 - exposing a report region, [17-27](#)
 - specifying proxy server address, [17-10](#)
 - SSL enabled, [17-10](#)
- testing a RESTful Web service reference, [17-16](#)
- testing Web service reference manually, [17-14](#)
- using, [17-9](#)
- viewing Web service reference history, [17-27](#)
- Web service reference repository, [17-17](#)
- Web Services References page, [17-9](#)

Websheet applications

- about, [2-3](#), [5-1](#)
- access control list, [5-12](#)
- controlling access, [5-12](#)
- copying a page, [5-4](#)
- creating, [5-1](#)

- Websheet applications (*continued*)
 - creating access control list entries, [5-14](#)
 - creating from App Builder, [5-2](#)
 - creating from running Websheet, [5-3](#)
 - deleting, [5-7](#)
 - editing page details, [5-6](#)
 - monitoring activity, [5-7](#)
 - running, [5-4](#)
 - specifying access control list, [5-13](#)
 - understanding Websheet properties, [5-8](#)
 - viewing Websheet dashboard, [5-7](#)
- Websheet authentication
 - Application Express Account, [5-11](#)
 - Custom, [5-11](#)
 - LDAP, [5-11](#)
 - Single Sign-On, [5-11](#)
- Websheet dashboard, viewing, [5-7](#)
- Websheet properties
 - about, [5-9](#)
 - Access Control List Type, [5-12](#)
 - Allow Public Access, [5-12](#)
 - Application Date Format, [5-9](#)
 - Authentication, [5-11](#)
 - Cookie Domain, [5-11](#)
 - Cookie Name, [5-11](#)
 - Cookie Path, [5-11](#)
 - Default Application Language, [5-10](#)
 - Default Application Territory, [5-10](#)
 - Description, [5-10](#)
 - editing, [5-8](#)
 - editing SQL access, [5-12](#)
 - Home Page, [5-9](#)
 - Login Message, [5-10](#)
 - Logo, [5-10](#)
 - Logo Type, [5-10](#)
 - Logout URL, [5-11](#)
 - Name, [5-9](#)
 - Owner, [5-10](#)
 - Secure, [5-11](#)
 - Show Reset Password, [5-10](#)
 - Status, [5-9](#)
 - Style, [5-10](#)
 - Websheet Email From Address, [5-10](#)
- Websheets
 - about navigation, [3-21](#)
 - editing, [3-28](#)
 - modifying, [3-28](#)
 - navigating, [3-20](#)
 - understanding navigation, [3-22](#)
 - using breadcrumbs, [3-20](#)
- wizards
 - Create Page, [6-45](#)
 - creating a branch, [14-27](#)
 - creating a calendar, [10-1](#)
 - creating a chart, [10-23](#)
- wizards (*continued*)
 - creating a legacy chart, [24-32](#)
 - creating a list, [16-29](#), [16-35](#)
 - creating a LOV, [16-25](#)
 - creating a page computation, [15-2](#), [15-5](#), [17-24](#)
 - creating a page process, [15-27](#)
 - creating a region, [12-18](#)
 - creating an application, [2-3](#)
 - creating applications, [2-1](#)
 - creating navigation bar entry, [10-39](#), [14-21–14-23](#)
 - creating validations, [15-12](#), [15-15–15-17](#)
- workspace
 - searching, [1-7](#)
- Workspace Activity reports
 - Application Errors, [21-5](#)
 - Developer Activity, [21-5](#)
 - Environment, [21-5](#)
 - Login Attempts, [21-5](#)
 - Page View Analysis, [21-5](#)
 - Page Views, [21-5](#)
 - Sessions, [21-5](#)
 - Workspace Schema Reports, [21-5](#)
- Workspace administrator, [19-2](#)
- Workspace home page
 - Accessibility Mode region, [1-11](#)
 - Administration menu, [1-8](#)
 - Available Updates region, [1-11](#)
 - News entries, [1-5](#)
 - system message, [1-5](#)
 - Top Apps region, [1-5](#)
 - Top Users region, [1-5](#)
- workspace news
 - managing, [4-28](#)
- workspace provisioning, enabling self service, [19-11](#)
- workspace settings, configuring, [19-12](#)
- workspace themes
 - adding, [11-26](#)
 - deleting, [11-27](#)
 - exporting, [11-28](#)
 - managing, [11-25](#), [11-28](#)
 - modifying, [11-27](#)
 - viewing existing, [11-27](#)
- workspace to schema assignments, managing, [19-13](#)
- Workspace Utilities
 - App Builder Defaults, [1-16](#)
 - Application Express Views, [1-16](#)
 - Application Groups, [1-16](#)
 - Cross Application Reports, [1-16](#)
 - Export, [1-16](#)
 - Manage News, [1-16](#)
 - Workspace Themes, [1-16](#)

workspaces, managing, [19-12](#)
WSDL document, [17-9](#)

X

XLIFF, [20-9](#)
 applying, [20-12](#)
 deleting an uploaded document, [20-13](#)
 Target Elements, [20-11](#)

XLIFF (*continued*)
 uploading, [20-12](#)
 uploading and publishing, [20-12](#)

Z

Zip file
 downloading static application files, [16-15](#)
 downloading static workspace files, [16-19](#)