# Contents

**Preface** ......................................................................................................................................................... xxv  
  Audience .............................................................................................................................................. xxv  
  Documentation Accessibility ............................................................................................................... xxv  
  Related Documents ........................................................................................................................... xxv  
  Conventions ............................................................................................................................................. xxvi  

**Changes in This Release** ........................................................................................................................... xxvii  
  Changes in Oracle Application Express Release 5.0 ........................................................................ xxvii  

1 Quick Start  

Understanding Oracle Application Express ................................................................................................. 1-1  
  What is Oracle Application Express? ........................................................................................................ 1-1  
  How Oracle Application Express Works ..................................................................................................... 1-2  
  About the Differences Between Runtime and Full Development Environments .................................. 1-2  

Signing In To Oracle Application Express .................................................................................................. 1-2  
  Browser Requirements ............................................................................................................................. 1-3  
  About Application Express User Roles .................................................................................................... 1-3  
  Overview of Configuring Oracle Application Express .......................................................................... 1-3  
  Signing In and Out of a Workspace as a Developer ............................................................................... 1-6  

Understanding the Workspace Home Page .................................................................................................. 1-10  
  Workspace Home Page ............................................................................................................................ 1-11  
  Top Applications and Top Users Dashboards ......................................................................................... 1-12  
  News and Messages ............................................................................................................................... 1-12  
  Developer Navigation Tools ....................................................................................................................... 1-13  
  Available Updates Region ......................................................................................................................... 1-16  
  Accessibility Mode Region ....................................................................................................................... 1-17  

Editing Your Account Profile ......................................................................................................................... 1-17  
  Editing Your Account Profile and Preferences ...................................................................................... 1-17  
  Changing Your Password .......................................................................................................................... 1-18  

Editing User Preferences ............................................................................................................................... 1-18  

Using Oracle Application Express Documentation ..................................................................................... 1-19  
  Accessing Oracle Help Center ............................................................................................................... 1-19  
  Searching the Oracle Help Center ......................................................................................................... 1-20  
  Downloading Online Documents ............................................................................................................ 1-20  
  Searching a Downloaded Documentation Library ................................................................................... 1-21
### 2 Application Builder Concepts

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding Oracle Application Express Applications</td>
<td>2-1</td>
</tr>
<tr>
<td>What is an Oracle Application Express Application?</td>
<td>2-1</td>
</tr>
<tr>
<td>What is the Difference Between a Database and Websheet Application?</td>
<td>2-2</td>
</tr>
<tr>
<td>About Running the Create Application Wizard</td>
<td>2-2</td>
</tr>
<tr>
<td>Understanding Application Pages</td>
<td>2-4</td>
</tr>
<tr>
<td>What Is an Application Page?</td>
<td>2-4</td>
</tr>
<tr>
<td>Understanding the Difference Between Page Designer and Component View</td>
<td>2-4</td>
</tr>
<tr>
<td>Viewing a Page in Page Designer</td>
<td>2-6</td>
</tr>
<tr>
<td>Viewing a Page in Component View</td>
<td>2-7</td>
</tr>
<tr>
<td>Switching from Page Designer to Component View</td>
<td>2-7</td>
</tr>
<tr>
<td>Switching from Component View to Page Designer</td>
<td>2-8</td>
</tr>
<tr>
<td>Managing User Interface Design</td>
<td>2-8</td>
</tr>
<tr>
<td>About Determining Your Application User Interface</td>
<td>2-8</td>
</tr>
<tr>
<td>About Mobile Application Design</td>
<td>2-8</td>
</tr>
<tr>
<td>Understanding the Difference Between Mobile Design and Responsive Design</td>
<td>2-9</td>
</tr>
<tr>
<td>Running an Application or Page</td>
<td>2-9</td>
</tr>
<tr>
<td>Run Application and Run Page Buttons</td>
<td>2-10</td>
</tr>
<tr>
<td>How Your Browser Impacts the Way Applications Run</td>
<td>2-10</td>
</tr>
<tr>
<td>Running an Application from the Application Builder Home Page</td>
<td>2-10</td>
</tr>
<tr>
<td>Running an Application from the Application Home Page</td>
<td>2-11</td>
</tr>
<tr>
<td>Running a Page on the Application Home Page</td>
<td>2-11</td>
</tr>
<tr>
<td>Running a Page in Page Designer</td>
<td>2-11</td>
</tr>
<tr>
<td>Running a Page in Component View</td>
<td>2-11</td>
</tr>
<tr>
<td>How Does Page Processing and Page Rendering Work?</td>
<td>2-11</td>
</tr>
<tr>
<td>How the Application Express Engine Renders and Processes Pages</td>
<td>2-12</td>
</tr>
<tr>
<td>Understanding Conditional Rendering and Processing</td>
<td>2-12</td>
</tr>
<tr>
<td>About Verifying User Identity</td>
<td>2-13</td>
</tr>
<tr>
<td>About Controlling Access to Controls and Components</td>
<td>2-14</td>
</tr>
<tr>
<td>Understanding Session State Management</td>
<td>2-14</td>
</tr>
<tr>
<td>What is Session State?</td>
<td>2-14</td>
</tr>
<tr>
<td>What Is a Session?</td>
<td>2-14</td>
</tr>
<tr>
<td>About Session IDs</td>
<td>2-15</td>
</tr>
<tr>
<td>Viewing Session State</td>
<td>2-15</td>
</tr>
<tr>
<td>Understanding URL Syntax</td>
<td>2-16</td>
</tr>
<tr>
<td>About the Application Page URL</td>
<td>2-17</td>
</tr>
<tr>
<td>About Using ( \text{f?p} ) Syntax to Link Pages</td>
<td>2-17</td>
</tr>
<tr>
<td>Application Alias</td>
<td>2-19</td>
</tr>
<tr>
<td>Page Alias</td>
<td>2-19</td>
</tr>
<tr>
<td>Calling a Page Using an Application and Page Alias</td>
<td>2-19</td>
</tr>
<tr>
<td>Calling a Page from a Button URL</td>
<td>2-20</td>
</tr>
<tr>
<td>About Enabling Support for Bookmarks</td>
<td>2-20</td>
</tr>
<tr>
<td>Managing Session State Values</td>
<td>2-20</td>
</tr>
<tr>
<td>About Referencing Session State</td>
<td>2-20</td>
</tr>
<tr>
<td>Setting Session State</td>
<td>2-21</td>
</tr>
</tbody>
</table>
5 Managing the Application Life Cycle with Team Development

Accessing Team Development ................................................................. 5-1
Team Development Home Page ............................................................... 5-1
About Utilizing the Tags Summary .......................................................... 5-3
Tracking Features .................................................................................. 5-3
  Viewing the Features Report ................................................................. 5-3
  Creating a Feature ............................................................................... 5-4
  About Creating Parent Features ......................................................... 5-4
  Updating a Feature ............................................................................ 5-4
  How to Attach Files to a Feature ......................................................... 5-4
  Viewing Other Feature Reports .......................................................... 5-5
Tracking Milestones .............................................................................. 5-8
  Viewing the Milestones Report ........................................................... 5-8
  Creating a Milestone .......................................................................... 5-8
  Updating a Milestone ........................................................................ 5-9
  Viewing Other Milestone Reports ....................................................... 5-9
Tracking To Dos .................................................................................... 5-10
  Viewing To Dos ................................................................................ 5-10
  Creating a To Do ............................................................................... 5-11
  Updating a To Do ............................................................................. 5-11
  How to Attach Files to a To Do .......................................................... 5-11
  Viewing Other To Do Reports ............................................................ 5-12
Managing Bugs ..................................................................................... 5-13
  Viewing Bugs .................................................................................. 5-13
  Creating a Bug .................................................................................. 5-13
  Editing a Bug .................................................................................... 5-14
  How to Attach Files to a Bug .............................................................. 5-14
  Viewing Other Bug Reports ............................................................... 5-15
Managing Feedback .............................................................................. 5-16
  Viewing Feedback ............................................................................ 5-16
  Adding a Feedback Page to an Application ....................................... 5-16
  Submitting Feedback ....................................................................... 5-17
  Reviewing Feedback ........................................................................ 5-18
  Viewing Other Feedback Reports ..................................................... 5-18
Managing Team Development Utilities ................................................. 5-19
  Accessing Team Development Utilities ............................................ 5-19
  Configuring Team Development Settings ........................................ 5-20
  Viewing a Release Summary .............................................................. 5-21
  Managing Feature Utilities ................................................................ 5-21
  Editing Focus Areas ......................................................................... 5-21
  Viewing File Attachments ................................................................ 5-22
## 6 Creating Websheet Applications

**About Websheet Applications** ............................................................................................ 6-1

**Creating a Websheet Application** ........................................................................................ 6-1
- Creating a Websheet from Application Builder ................................................................. 6-2
- Creating a Websheet Application from a Running Websheet ............................................. 6-3

**Copying an Existing Websheet Page** .................................................................................. 6-3

**Running a Websheet** ........................................................................................................... 6-3
- Running a Websheet from the Application Builder Home Page ........................................ 6-4
- Running a Websheet from the Application Home Page ...................................................... 6-4
- Determining a Websheet URL .............................................................................................. 6-5

**Editing Page Details** .......................................................................................................... 6-5

**Deleting a Websheet Application** ....................................................................................... 6-5

**Viewing the Websheet Dashboard** ..................................................................................... 6-5

**Monitoring Websheet Activity** ........................................................................................... 6-6

**Understanding Websheet Properties** ................................................................................ 6-6
- Editing Websheet Properties ................................................................................................. 6-7
- Websheet Properties ............................................................................................................ 6-7

**Controlling Websheet Access** ............................................................................................ 6-10
- How Websheet Access Works ............................................................................................... 6-10
- Specifying the Access Control List Type .............................................................................. 6-11
- Creating Access Control List Entries ................................................................................... 6-11

## 7 Creating Database Applications

**Using the Create Application Wizard** .................................................................................. 7-1
- About the Create Application Wizard .................................................................................. 7-2
- Available Page Types When Running the Create Application Wizard ................................ 7-2

**Creating a Database Application** .......................................................................................... 7-4
- Creating a Database Application Based on a Table, Query, or Drill-Down Query ................. 7-4
- Creating a Desktop Database Application from a Spreadsheet .......................................... 7-7
- Copying a Database Application ............................................................................................ 7-8

**Creating Mobile Applications** ........................................................................................... 7-10
- About jQuery Mobile Support ............................................................................................... 7-10
- Creating a Mobile Application .............................................................................................. 7-10

**Deleting an Application** ....................................................................................................... 7-13
- Deleting an Application from the Application Home Page .................................................. 7-13
- Deleting an Application from the Edit Application Definition Page .................................... 7-13

**Managing Application Attributes** ...................................................................................... 7-13
- Editing the Application Attributes ....................................................................................... 7-14
- Configuring Security Attributes ............................................................................................ 7-19
- Configuring Globalization Attributes .................................................................................. 7-26
- Managing the Application User Interface ............................................................................. 7-29

**Managing Pages in a Database Application** ...................................................................... 7-36
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Adding Pages to a Database Application</td>
<td>7-36</td>
</tr>
<tr>
<td>Available Pages Types When Running the Create Page Wizard</td>
<td>7-36</td>
</tr>
<tr>
<td>Creating a Page from the Application Home Page</td>
<td>7-40</td>
</tr>
<tr>
<td>Creating Dialog Pages</td>
<td>7-40</td>
</tr>
<tr>
<td>About Managing Pages in Page Designer</td>
<td>7-43</td>
</tr>
<tr>
<td>About Managing Pages in Component View</td>
<td>7-44</td>
</tr>
<tr>
<td>About Copying Database Application Pages</td>
<td>7-44</td>
</tr>
<tr>
<td>About Copying Pages Between User Interfaces</td>
<td>7-44</td>
</tr>
<tr>
<td>About Deleting Database Application Pages</td>
<td>7-45</td>
</tr>
<tr>
<td><strong>Using the Find Icon</strong></td>
<td>7-45</td>
</tr>
<tr>
<td>About the Find Icon</td>
<td>7-45</td>
</tr>
<tr>
<td>Using the Find Icon</td>
<td>7-45</td>
</tr>
<tr>
<td>Searching for Items</td>
<td>7-46</td>
</tr>
<tr>
<td>Searching for Pages</td>
<td>7-47</td>
</tr>
<tr>
<td>Searching for Queries</td>
<td>7-47</td>
</tr>
<tr>
<td>Searching for Tables</td>
<td>7-48</td>
</tr>
<tr>
<td>Using the PL/SQL Finder</td>
<td>7-49</td>
</tr>
<tr>
<td>Searching for Images</td>
<td>7-49</td>
</tr>
<tr>
<td><strong>Creating Page Groups</strong></td>
<td>7-50</td>
</tr>
<tr>
<td>Viewing Page Groups</td>
<td>7-50</td>
</tr>
<tr>
<td>Creating a Page Group</td>
<td>7-51</td>
</tr>
<tr>
<td>Editing a Page Group Definition</td>
<td>7-51</td>
</tr>
<tr>
<td>Assigning a Page to a Page Group</td>
<td>7-51</td>
</tr>
<tr>
<td>Viewing Pages by Page Group</td>
<td>7-52</td>
</tr>
<tr>
<td>Deleting a Page Group</td>
<td>7-52</td>
</tr>
<tr>
<td><strong>Locking and Unlocking a Page</strong></td>
<td>7-52</td>
</tr>
<tr>
<td>Determining If a Page Is Locked</td>
<td>7-53</td>
</tr>
<tr>
<td>Locking a Page</td>
<td>7-53</td>
</tr>
<tr>
<td>Unlocking a Page</td>
<td>7-54</td>
</tr>
<tr>
<td><strong>Adding Developer Comments</strong></td>
<td>7-56</td>
</tr>
<tr>
<td>Developer Comment, Bug, or To Do Button</td>
<td>7-56</td>
</tr>
<tr>
<td>Adding Developer Comments to an Application</td>
<td>7-56</td>
</tr>
<tr>
<td>Viewing and Editing Developer Comments</td>
<td>7-56</td>
</tr>
<tr>
<td>Deleting Developer Comments</td>
<td>7-57</td>
</tr>
<tr>
<td>Accessing the Developer Comments from Shared Components</td>
<td>7-58</td>
</tr>
<tr>
<td>Viewing the Developer Comments Calendar</td>
<td>7-58</td>
</tr>
<tr>
<td><strong>Adding Database Application Comments</strong></td>
<td>7-58</td>
</tr>
<tr>
<td><strong>Managing Application Groups</strong></td>
<td>7-59</td>
</tr>
<tr>
<td>Viewing Database Application Groups</td>
<td>7-59</td>
</tr>
<tr>
<td>Creating a Database Application Group</td>
<td>7-60</td>
</tr>
<tr>
<td>Assigning a Database Application to an Application Group</td>
<td>7-60</td>
</tr>
<tr>
<td>Removing a Database Application from an Application Group</td>
<td>7-60</td>
</tr>
<tr>
<td>Deleting an Database Application Group</td>
<td>7-60</td>
</tr>
<tr>
<td><strong>Accessing Application Utilities</strong></td>
<td>7-61</td>
</tr>
<tr>
<td>Using Application Builder Utilities</td>
<td>7-61</td>
</tr>
<tr>
<td>Accessing Page Specific Utilities</td>
<td>7-65</td>
</tr>
<tr>
<td>Utilizing Cross Page Utilities</td>
<td>7-66</td>
</tr>
</tbody>
</table>
8 Managing Pages in Page Designer

About the Page Designer Window ........................................................................................................ 8-1
Understanding Common Page Designer UI Elements ........................................................................... 8-2
  Page Designer Toolbar .......................................................................................................................... 8-3
  Tree Pane ........................................................................................................................................... 8-5
  Central Pane ....................................................................................................................................... 8-6
  Grid Layout .......................................................................................................................................... 8-7
  Gallery ................................................................................................................................................ 8-9
  Messages ............................................................................................................................................ 8-9
  Page Search ......................................................................................................................................... 8-10
  Help .................................................................................................................................................. 8-11
  Property Editor ................................................................................................................................. 8-11
  Code Editor ......................................................................................................................................... 8-12
Using Page Designer .................................................................................................................................. 8-14
  Viewing a Page in Page Designer ........................................................................................................ 8-15
  Accessing the Property Editor ............................................................................................................ 8-15
  Adding a Region, Item, or Button by Dragging and Dropping .............................................................. 8-15
  Adding a Region, Item, or Button Using Gallery Context Menus ........................................................... 8-16
  Adding a Region, Item, or Button from the Rendering Tree ................................................................. 8-17
  Creating Pre-Rendering Branches, Computations, and Processes ..................................................... 8-17
  Creating Page Submission Branches, Validations, Computations, and Processes ............................. 8-18
  Searching for Pages ............................................................................................................................ 8-18
Editing Page Attributes in Page Designer ............................................................................................. 8-19
  Viewing Page Attributes in the Property Editor .................................................................................. 8-19
  Property Editor - Page ....................................................................................................................... 8-20
  Incorporating JavaScript Using Page Designer .................................................................................... 8-20
  Configuring Rejoin Sessions in Page Designer .................................................................................... 8-21
  Configuring Dialogs Attributes in Page Designer ................................................................................ 8-22
Running a Page in Page Designer ........................................................................................................ 8-24
Runtime Developer Toolbar .................................................................................................................. 8-24
Copying a Database Application Page in Page Designer ..................................................................... 8-25
Deleting Pages in Page Designer .......................................................................................................... 8-26
  Deleting a Page in Page Designer ........................................................................................................ 8-26
  Deleting Multiple Pages ...................................................................................................................... 8-26
  Deleting a Range of Pages ................................................................................................................... 8-27

9 Managing Pages in Component View

About the Component View Window .................................................................................................... 9-2
Understanding Component View Common UI Elements ....................................................................... 9-2
  Component View Toolbar ................................................................................................................... 9-3
  Navigation Bar .................................................................................................................................... 9-3
Using Component View ......................................................................................................................... 9-5
  Viewing a Page in Component View .................................................................................................. 9-5
Editing a Page in Component View ................................................................. 9-5
Understanding Page Rendering ....................................................................... 9-7
Understanding Page Processing ...................................................................... 9-8
Understanding Shared Components ................................................................ 9-9
Creating a Page Control Component View ..................................................... 9-10

**Editing Page Attributes in Component View** ............................................... 9-11
Viewing Page Attributes in Component View .................................................. 9-11
Page Attributes in Component View ............................................................... 9-11
Incorporating JavaScript Using Component View ........................................... 9-21
Configuring Rejoin Sessions in Component View ........................................... 9-22
Configuring Dialog Attributes in Component View ......................................... 9-23

**Running an Application or Page in Component View** ............................... 9-24

**Runtime Developer Toolbar** ....................................................................... 9-25

**Copying a Database Application Page in Component View** ...................... 9-26

**Deleting Pages in Component View** ............................................................. 9-26
Deleting a Page in Component View ............................................................... 9-26
Deleting a Page While Editing Page Attributes .............................................. 9-26
Deleting Multiple Pages .................................................................................. 9-27
Deleting a Range of Pages .............................................................................. 9-27

**10 Developing Reports**

**Understanding the Difference Between Interactive and Classic Reports** .... 10-1
Interactive Reports ........................................................................................... 10-1
Classic Reports .................................................................................................. 10-2

**Creating Reports Using a Wizard** ............................................................... 10-3
Create Application Wizard Report Options .................................................... 10-4
Create Page Wizard Report Options ................................................................ 10-5
Create Region Wizard Report Options ............................................................ 10-6
Creating a Report Using the Create Application Wizard .................................. 10-8
Creating a Report Using the Create Page Wizard ............................................ 10-9
Creating a Report in Page Designer Using a Wizard ........................................ 10-9

**Creating a Report by Dragging and Dropping from the Gallery** .................. 10-9

**Editing Interactive Reports in Page Designer** ........................................... 10-11
Editing Region Attributes in Page Designer .................................................... 10-11
Editing Column Attributes in Page Designer .................................................. 10-12
Editing Interactive Report Attributes in Page Designer .................................... 10-13
Controlling Interactive Report Pagination in Page Designer .......................... 10-13
Customizing the Interactive Report Search Bar ............................................. 10-14
Customizing the Actions Menu in Page Designer ............................................ 10-16
Managing Link Columns in Page Designer ...................................................... 10-20
Creating a Column Link in an Interactive Report ........................................... 10-22
Displaying Interactive Reports Conditionally ................................................ 10-23
Defining a Column as a List of Values in an Interactive Report ....................... 10-24
Enabling Icon View in Page Designer ............................................................. 10-24
Formatting Columns in an Interactive Report ................................................. 10-26
Enabling Detail View in Page Designer ............................................................ 10-26
Configuring Advanced Attributes for Interactive Reports .............................. 10-27
Editing Interactive Reports in Component View ................................................................. 10-33
  Editing Interactive Report Attributes in Component View ............................................... 10-33
  Editing Column Attributes in Component View ............................................................... 10-35
  Creating Column Groups ..................................................................................................... 10-36
  Controlling Interactive Report Pagination in Component View ...................................... 10-36
  Managing the Search Bar and Actions Menu in Component View .................................... 10-37
  Managing Link Columns in Component View .................................................................... 10-43
  Creating a Column Link in an Interactive Report ............................................................. 10-45
  Displaying Interactive Reports Conditionally ..................................................................... 10-47
  Defining a Column as a List of Values in an Interactive Report ........................................ 10-47
  Formatting Columns in an Interactive Report .................................................................... 10-48
  Enabling Icon View in Component View ........................................................................... 10-48
  Enabling Detail View in Component View .......................................................................... 10-49
  Configuring Interactive Report Advanced Attributes ........................................................ 10-50
  Linking to Interactive Reports .......................................................................................... 10-52
  Linking to Shared Interactive Reports ............................................................................... 10-55
  Editing Saved Default Interactive Reports in Component View ....................................... 10-56
  Viewing Saved Interactive Report Activity ........................................................................ 10-56

Customizing Interactive Reports as an End User ................................................................. 10-57
  About the Search Bar .......................................................................................................... 10-57
  About the Actions Menu ..................................................................................................... 10-58
  Using Interactive Report Filters ......................................................................................... 10-59
  Saving Interactive Reports .................................................................................................. 10-60

Editing Classic Reports in Page Designer ............................................................................ 10-64
  Editing Classic Report Attributes in Page Designer .......................................................... 10-65
  Altering Classic Report Layout in Page Designer ............................................................... 10-66
  Controlling Classic Report Pagination in Page Designer ................................................ 10-67
  Enabling Column Sorting in a Classic Report in Page Designer ....................................... 10-68
  Adding a Download Link to a Classic Report in Page Designer ........................................ 10-68
  Enabling Classic Report CSV Output Option in Page Designer ....................................... 10-68
  Adding Support for XML File or a CSV File Export in Page Designer ............................. 10-69
  Creating a Column Link in a Classic Report in Page Designer .......................................... 10-69
  Defining an Updatable Column in a Classic Report in Page Designer ............................. 10-70
  Defining a Column as a List of Values in a Classic Report in Page Designer ..................... 10-71
  Controlling When Classic Report Columns Display in Page Designer ............................ 10-71
  Controlling Classic Report Column Breaks in Page Designer .......................................... 10-72

Editing Classic Reports in Component View ........................................................................ 10-72
  Editing Classic Report Attributes in Component View .................................................... 10-73
  Altering Classic Report Layout in Component View ......................................................... 10-74
  Controlling Classic Report Pagination in Component View .......................................... 10-75
  Enabling Column Sorting in a Classic Report in Component View ................................ 10-77
  Adding a Download Link to a Classic Report in Component View .................................. 10-78
  Enabling Classic Report CSV Output in Component View .............................................. 10-78
  Adding Support for XML File or a CSV File Export ......................................................... 10-78
  Creating a Column Link in a Classic Report in Component View .................................... 10-79
  Defining an Updatable Column in Component View ......................................................... 10-80
11 Developing Forms

Using a Wizard to Create a Form .......................................................... 11-1
Create Page Wizard Form Options .................................................... 11-2
Create Region Wizard Form Options ................................................ 11-3
Creating a Form Using the Create Page Wizard ............................... 11-4
Creating a Form Using the Create Application Wizard ................... 11-5
Creating a Form in Page Designer .................................................... 11-6

Managing Tabular Forms ................................................................. 11-6
About Creating Tabular Forms ........................................................ 11-6
About Running a Tabular Form ........................................................ 11-7
How Tabular Forms Work ............................................................... 11-8
Creating a Tabular Form Using the Create Application Wizard ....... 11-8
Creating a Tabular Form Using the Create Page Wizard ................. 11-9
Synchronizing Tabular Form Column Order with Region Source ....... 11-12

Managing Master Detail Forms .......................................................... 11-12
About Master Detail Forms .............................................................. 11-12
Creating a Master Detail Form Using the Create Application Wizard . 11-14
Creating a Master Detail Form Using the Create Page Wizard .......... 11-15

Creating a Form Manually in Component View ............................... 11-18
About Creating Forms Manually in Component View ...................... 11-18
Processing a Form ........................................................................ 11-18
Populating Forms in Component View ............................................ 11-20

Validating User Input in Forms ......................................................... 11-21
About Creating Validations and Error Messages .............................. 11-21
Associating a Validation with a Specific Item in Page Designer ......... 11-21
Associating a Validation with a Specific Item in Component View .... 11-21
About the Error Message Display Location ..................................... 11-22

Understanding BLOB Support in Forms ............................................ 11-22
About BLOB Support in Forms ......................................................... 11-22
Managing Database Application Components

Creating Calendars
- About Supported Calendars
- Sample Calendar
- Adding a Calendar to a New Page
- Managing Calendar Attributes in Page Designer
- Managing Calendars in Component View
- About the Date Column Format Mask
- Enabling the Dragging and Dropping of Data In an Existing Calendar
- About Supported Calendar Substitution Strings in Legacy Calendars

Creating Maps
- About Map Chart Support
- About Creating SQL Queries for Maps
- Adding Map Charts to a New Page
- Managing Map Charts in Page Designer
- Managing Map Charts in Component View

Creating Charts
- About Chart Types
- Creating SQL Queries for Charts
- Adding a Chart to a New Page
- Managing Charts in Page Designer
- Managing Charts in Component View
- Understanding SVG Chart Support
- Understanding HTML Chart Support
- About Saving Flash Charts
- Using Custom XML with Flash and HTML5 Charts

Creating Help for Your Application
- Creating Help in Page Designer
- Creating Help in Component View
- Editing Multiple Item Help Topics Simultaneously
- Creating a Help Navigation Bar Entry

Using Themes and Theme Styles

Using Themes
- About Themes
- About Responsive Design
- About Grid-based Themes
- Using Theme Styles
- Understanding Template Options
- Accessing the Themes Page
- Editing Themes

Managing Themes and Subscriptions
### About Regions

- Creating a Global Page to Display Components on Every Page
- Optimizing a Page for Printing

### Understanding Theme Subscriptions

- Creating a Theme
- Switching Themes
- Changing a Theme Identification Number
- Copying a Theme
- Deleting a Theme
- Creating a Master Theme Application
- Managing Workspace Themes
- Managing Instance Themes
- About Exporting and Importing Themes
- Viewing Theme Reports

### Creating Custom Themes

- About Cascading Style Sheets
- About Calling the JavaScript File from the Page Template
- About Using Escaping Syntax in Substitution Strings
- Selecting a Default Page Template
- Creating a New Template
- Using Template Options
- Viewing Template Reports
- Managing Templates
- Breadcrumb Templates
- Button Templates
- Legacy Calendar Templates
- Label Templates
- List Templates
- Page Templates
- Popup LOV Templates
- Region Templates
- Report Templates

### Using Custom Cascading Style Sheets

- About Uploading Cascading Style Sheets
- Referencing an Uploaded Cascading Style Sheet in the Page Template

### Controlling Page Layout

- About Page Layout in Oracle Application Express
- Optimizing a Page for Printing
- Selecting a Printer Friendly Template for an Application
- Using f?p Syntax to Toggle to Printer Friendly Mode
- Creating a Global Page to Display Components on Every Page
- Creating a Global Page
- Navigating to the Global Page

### About Regions

- Managing Regions in Page Designer
- Creating a Region in Page Designer
- Managing Regions in Page Designer
- Copying a Region in Page Designer
- Deleting Regions in Page Designer
16  Adding Navigation

About Creating Lists as Navigation ....................................................................................... 16-1
Creating Tabs .......................................................................................................................... 16-1
   About Tabs ............................................................................................................................ 16-2
   About the Tabs Page ................................................................................................................ 16-2
   Checking for Tab Template Support ....................................................................................... 16-3
   Creating a Tab ......................................................................................................................... 16-4
   Editing Tabs ............................................................................................................................. 16-5
   Accessing Tab Reports ............................................................................................................ 16-6
Creating Breadcrumbs ............................................................................................................. 16-7
   What Are Breadcrumbs? .......................................................................................................... 16-7
   Creating a Breadcrumb While Creating a Page ....................................................................... 16-8
   Creating a Breadcrumb Manually ............................................................................................. 16-8
   Editing Breadcrumbs ............................................................................................................. 16-13
   Reparenting Breadcrumb Entries ........................................................................................... 16-14
   Deleting Unused Breadcrumb Entries .................................................................................... 16-14
   Accessing Breadcrumb Reports .............................................................................................. 16-15
Creating a Navigation Bar Entry ............................................................................................. 16-15
   About Navigation Bars ........................................................................................................... 16-16
   Creating a Navigation Bar Entry ............................................................................................. 16-16
   Copying a Navigation Bar Entry ............................................................................................. 16-18
   Editing a Navigation Bar Entry ............................................................................................... 16-19
   Editing Multiple Navigation Bar Entries Simultaneously ..................................................... 16-19
   Accessing Navigation Bar Entry Reports ............................................................................. 16-20
Controlling Navigation Using Branches ................................................................................ 16-20
   Creating a Branch in Page Designer ....................................................................................... 16-20
   Creating a Branch in Component View .................................................................................. 16-21
   About Branching Conditionally ............................................................................................ 16-22

17  Managing Computations, Validations, and Processes

Understanding Page Computations ......................................................................................... 17-1
   About Page Computations ..................................................................................................... 17-1
   Managing Page Computations in Page Designer .................................................................. 17-2
   Managing Page Computations in Component View ............................................................... 17-3
   Computation Points and Computation Syntax Example ....................................................... 17-5
Understanding Validations ...................................................................................................... 17-6
   About Validations .................................................................................................................. 17-7
   Managing Validations in Page Designer ................................................................................ 17-7
   Managing Validations in Component View ............................................................................ 17-8
   About Determining When Validations Execute .................................................................... 17-15
   About Defining How Validation Error Messages Display .................................................. 17-16
   Processing Validations Conditionally .................................................................................... 17-16
   About Altering the Go to Error Link ....................................................................................... 17-17
   What Happens When a Validation Fails? ............................................................................ 17-17
Understanding Page Processes ............................................................................................... 17-17
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Page Processes</td>
<td>17-18</td>
</tr>
<tr>
<td>Managing Page Processes in Page Designer</td>
<td>17-18</td>
</tr>
<tr>
<td>Managing Page Processes in Component View</td>
<td>17-21</td>
</tr>
<tr>
<td>About Branches</td>
<td>17-25</td>
</tr>
<tr>
<td>Understanding Application Processes</td>
<td>17-30</td>
</tr>
<tr>
<td>About On Demand Application Processes</td>
<td>17-26</td>
</tr>
<tr>
<td>About Application Processes that Execute On New Instance</td>
<td>17-26</td>
</tr>
<tr>
<td>About Running an On Demand Process from a Page Request</td>
<td>17-26</td>
</tr>
<tr>
<td>Application Process Example</td>
<td>17-27</td>
</tr>
<tr>
<td>Creating an Application Process</td>
<td>17-27</td>
</tr>
<tr>
<td>Editing Application Process Attributes</td>
<td>17-28</td>
</tr>
<tr>
<td>Creating Application Process Error Messages</td>
<td>17-28</td>
</tr>
<tr>
<td>Viewing the Application Processes History Report</td>
<td>17-29</td>
</tr>
<tr>
<td>Understanding Application Computations</td>
<td>17-30</td>
</tr>
<tr>
<td>About Application Computations</td>
<td>17-30</td>
</tr>
<tr>
<td>About Application Computations that Execute On New Instance</td>
<td>17-30</td>
</tr>
<tr>
<td>Creating an Application Computation</td>
<td>17-31</td>
</tr>
<tr>
<td>Accessing the Application Computation History Report</td>
<td>17-31</td>
</tr>
<tr>
<td>Using the Attribute Dictionary</td>
<td>17-32</td>
</tr>
<tr>
<td>About the Attribute Dictionary</td>
<td>17-32</td>
</tr>
<tr>
<td>Accessing the Manage Attribute Dictionary</td>
<td>17-32</td>
</tr>
</tbody>
</table>

### 18 Managing Shared Components

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working with Shared Components</td>
<td>18-1</td>
</tr>
<tr>
<td>About the Advantages of Using Shared Components</td>
<td>18-1</td>
</tr>
<tr>
<td>Accessing the Shared Components Page</td>
<td>18-2</td>
</tr>
<tr>
<td>About the Shared Components Icon</td>
<td>18-2</td>
</tr>
<tr>
<td>Shared Components Page</td>
<td>18-7</td>
</tr>
<tr>
<td>Managing Component Settings</td>
<td>18-7</td>
</tr>
<tr>
<td>Editing Component Settings</td>
<td>18-7</td>
</tr>
<tr>
<td>Viewing the Component Settings History Report</td>
<td>18-8</td>
</tr>
<tr>
<td>Managing Static Application Files</td>
<td>18-8</td>
</tr>
<tr>
<td>Specifying the Location of Static Application Files</td>
<td>18-9</td>
</tr>
<tr>
<td>Uploading Static Application Files</td>
<td>18-9</td>
</tr>
<tr>
<td>Viewing the Static Application File Report</td>
<td>18-10</td>
</tr>
<tr>
<td>About Referencing Static Application Files</td>
<td>18-10</td>
</tr>
<tr>
<td>Downloading Static Application Files</td>
<td>18-10</td>
</tr>
<tr>
<td>Deleting Static Application Files</td>
<td>18-11</td>
</tr>
<tr>
<td>Managing Static Workspace Files</td>
<td>18-12</td>
</tr>
<tr>
<td>Uploading Static Workspace Files</td>
<td>18-12</td>
</tr>
<tr>
<td>Viewing Static Workspace File Report</td>
<td>18-13</td>
</tr>
<tr>
<td>About Referencing Static Workspace Files</td>
<td>18-13</td>
</tr>
<tr>
<td>Downloading Static Workspace Files</td>
<td>18-13</td>
</tr>
<tr>
<td>Deleting Static Workspace Files</td>
<td>18-14</td>
</tr>
<tr>
<td>Managing Application-Level Items</td>
<td>18-15</td>
</tr>
<tr>
<td>Creating an Application-Level Item</td>
<td>18-15</td>
</tr>
<tr>
<td>Viewing Application Item Usage</td>
<td>18-16</td>
</tr>
</tbody>
</table>
20 Managing Application Security

Understanding Administrator Security Best Practices .............................................. 20-1
  About Oracle Application Express Administrator Roles .................................... 20-2
  About Configuring Oracle REST Data Services with Oracle Application Express ........ 20-2
  About Configuring Oracle HTTP Server with mod_plsql with Oracle Application Express ....... 20-2
  About Security Considerations When Using the Embedded PL/SQL Gateway .......... 20-3
  Managing Instance Security .................................................................................. 20-3
  Configuring Instance Settings ............................................................................. 20-8
  About Configuring Workspace Purge Settings .................................................... 20-9
  Understanding Workspace Management .............................................................. 20-9
  About Integrating with Oracle BI Publisher ...................................................... 20-10
  About the Advantages of the Application Express Runtime Environment .......... 20-10
  Enabling Network Services in Oracle Database 11g or Later .............................. 20-10

Understanding Developer Security Best Practices .............................................. 20-14
  About Items of Type Password .......................................................................... 20-14
  Identifying At Risk Password Items ................................................................... 20-15
  Understanding Cross-Site Scripting Protection .................................................... 20-16
  About Session State and Security ...................................................................... 20-20
  Understanding Session State Protection ............................................................. 20-21
  About Securing File Uploads ............................................................................. 20-32

Controlling Access to Applications, Pages, and Page Components .................... 20-32
  How the Access Control List Works ................................................................... 20-32
  Creating an Access Control List ......................................................................... 20-33
  Selecting an Application Mode and Adding Users .............................................. 20-34
  About Controlling Access for Pages and Page Components ............................. 20-36

Establishing User Identity Through Authentication ............................................ 20-37
  Understanding Authentication ............................................................................ 20-37
  Creating an Authentication Scheme .................................................................... 20-39
  Understanding Preconfigured Authentication Schemes ....................................... 20-39
  Managing Existing Authentication Schemes .................................................... 20-47
  Viewing the Authentication Scheme Associated with an Application ................ 20-47
  Changing the Authentication Scheme Associated with an Application ............... 20-48
  Creating a Login Page ....................................................................................... 20-48
  Viewing Authentication Scheme Reports .......................................................... 20-48

Providing Security Through Authorization ......................................................... 20-49
  How Authorization Schemes Work .................................................................... 20-49
  Creating and Editing an Authorization Scheme ................................................ 20-49
21 Managing Application Globalization

Understanding Application Translation and Globalization Support ........................................ 21-1
   About the Translation Process ............................................................................................ 21-1
   About Language Identification ......................................................................................... 21-2
   About the Rules for Translating Database Applications ................................................. 21-2
   How Translated Applications Are Rendered ..................................................................... 21-2
   Making Application Attributes Translatable ..................................................................... 21-2

Specifying the Primary Language for an Application ............................................................ 21-4
   Using Format Masks for Items .......................................................................................... 21-5
   About Translating Applications for Multibyte Languages .............................................. 21-6

Understanding the Translation Process ............................................................................. 21-6
   Step 1: Map the Target Language .................................................................................... 21-6
   Step 2: Seed and Download to a Translation File ............................................................. 21-7
   Step 3: Translate the XLIFF File ....................................................................................... 21-9
   Step 4: Upload and Apply a Translated XLIFF Document ............................................... 21-10
   Step 5: Publish the Application ....................................................................................... 21-11
   Manually Editing a Translation ......................................................................................... 21-12

Translating Messages ........................................................................................................ 21-13
   About Translating Messages ............................................................................................ 21-13
   Translating Messages Used in PL/SQL Procedures .......................................................... 21-13
   Translating Messages Used Internally by Oracle Application Express ............................... 21-14

Translating Data That Supports List of Values .................................................................... 21-34
   Defining a Dynamic Translation ....................................................................................... 21-34

Understanding Supported Globalization Codes ............................................................... 21-34
   Oracle Application Express Globalization Codes ............................................................ 21-34

22 Debugging an Application

Running Advisor to Check Application Integrity ................................................................. 22-1
   About Oracle Application Express Advisor ...................................................................... 22-1
   Running Advisor on an Entire Application ...................................................................... 22-1
   Running Advisor on a Single Page .................................................................................. 22-2

Reviewing Session State ................................................................................................... 22-2

About Monitoring Application and Page Resource Use ...................................................... 22-3

Enabling SQL Tracing and Using TKPROF ......................................................................... 22-3

Utilizing Debug Mode ....................................................................................................... 22-3
   About Debug Mode ........................................................................................................... 22-4
   Enabling and Disabling Debug Mode for an Application .................................................. 22-4
   Running an Application in Debug Mode ........................................................................... 22-5
   Viewing Debug Reports in Development Mode ............................................................... 22-5
   About Using f?p Syntax to Access Debug Mode ................................................................. 22-5
   Viewing Debug Reports from a Running Application ...................................................... 22-6
   Viewing Debug Reports .................................................................................................... 22-6

Removing Controls and Components to Isolate a Problem ............................................... 22-7
23  Managing Application Performance

About Best Practices to Improve Performance ................................................................. 23-1
Identifying Performance Issues ................................................................................... 23-1
  About Troubleshooting Slow Running Applications .............................................. 23-2
  Diagnosing Performance Issues ............................................................................ 23-2
  About Debugging Problematic SQL Queries ......................................................... 23-2
  About Addressing Slow Queries ........................................................................... 23-2
Utilizing Logs and Reports ....................................................................................... 23-3
  How to Manage Log Files .................................................................................... 23-3
  Viewing the Application Dashboard Report .......................................................... 23-3
  Workspace Activity Reports .................................................................................. 23-4
  Instance Activity Reports ..................................................................................... 23-4
About Utilizing Database Reporting ................................................................. 23-5
About Database Parameters that Impact Performance ........................................... 23-5
About Limiting Resources ...................................................................................... 23-6
About Uploading Static Files to Your Web Server ............................................ 23-6
Creating Custom Activity Reports Using APEX_ACTIVITY_LOG ....................... 23-6
  Enabling Logging for an Application ................................................................ 23-6
  APEX_ACTIVITY_LOG ....................................................................................... 23-7
  APEX_ACTIVITY_LOG Sample Query ................................................................ 23-8

24  Deploying an Application

About Suggested Environments ............................................................................. 24-1
System Development Life Cycle Methodologies to Consider .................................. 24-2
  About Iterative Vs Planned Development ........................................................ 24-2
  About the Advantages of Creating Prototypes .................................................... 24-2
  Waterfall ................................................................................................................. 24-2
  Spiral ......................................................................................................................... 24-3
  Rapid Application Development ......................................................................... 24-3
Understanding the Deployment Process .............................................................. 24-3
  About Moving an Application to Another Instance ........................................ 24-3
  Deployment Options to Consider ................................................................. 24-4
  About Whether to Copy a Workspace ............................................................. 24-4
  About Whether to Copy a Database ................................................................. 24-4
  About the Application ID .................................................................................... 24-4
How to Move an Application to Another Instance ................................................ 24-5
  About Managing Database Objects .................................................................... 24-5
How to Create a Custom Packaged Application .................................................. 24-6
  How Creating a Custom Packaged Application Simplifies Deployment ........ 24-6
  Creating a Custom Packaged Application ...................................................... 24-7
  Adding an Access Control List to a Custom Packaged Application ............ 24-9
  Installing Supporting Objects .......................................................................... 24-10
  Deleting Supporting Objects Scripts, Messages, and Installation Options .... 24-10
  Upgrading a Custom Packaged Application .................................................... 24-10
  Deinstalling Supporting Objects ..................................................................... 24-12
  Viewing an Install Summary ............................................................................. 24-12
Oracle Application Express Application Builder User's Guide describes how to use the Oracle Application Express development environment to build and deploy database-centric web applications. Oracle Application Express turns a single Oracle database into a shared service by enabling multiple workgroups to build and access applications as if they were running in separate databases.

- **Audience**
- **Documentation Accessibility**
- **Related Documents**
- **Conventions**

**Audience**

Oracle Application Express Application 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**


**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](http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) or visit [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs](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
Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Changes in This Release

This preface contains:

- Changes in Oracle Application Express Release 5.0

Changes in Oracle Application Express Release 5.0

The following are changes in Oracle Application Express Application Builder User’s Guide for Oracle Application Express release 5.0.

New Features

The following features are new in this release:

- Page Designer
  
  Page Designer is a completely new Integrated Development Environment (IDE) designed to greatly improve developer productivity and quickly enhance and maintain pages within Oracle Application Express.

  - Cohesive User Experience
    
    No context-switching or drilling down when modifying multiple components. All modifications occur within Page Designer. Undo and redo changes as necessary, before saving the page.

  - Better Visual Representation
    
    Gain an immediate understanding of how your regions and items are laid out within the Grid Layout. Changes to components on a page are immediately reflected across the Tree and Grid Layout. Additionally, errors and warnings display immediately.

  - Intuitive Drag and Drop
    
    Page Designer provides an entirely new way to develop pages. In the Grid Layout pane, you can drag new components from the Component Gallery, or move and copy existing components around. In the Tree pane, you can drag to move and copy multiple components at a time.

  - Enhanced Code Editor
    
    A new code editor provides SQL and PL/SQL validation with inline errors, auto completion, syntax highlighting, search and replace with regex support, complete with undo and redo support.

See "Understanding Application Pages" on page 2-4 and "Managing Pages in Page Designer" on page 8-1.
■ New Design and User Interface Enhancements

Oracle Application Express features a user interface focused on improving the user experience through simplicity and removing clutter. The new design features a new color palette, icons, improved menus and navigation, better accessibility and keyboard support, more intuitive page layouts.

– Universal Theme

The Universal Theme is new application user interface that does away with excessive templates and supports effortless customizations with Theme Roller, Template Options, and Theme Styles. The Universal Theme empowers developers to build modern, responsive, sophisticated, accessible applications without requiring expert knowledge of HTML, CSS, or JavaScript.

To learn more about the Universal Theme, install the Universal Theme Sample Application. See "Installing a Packaged Application" on page 3-4.

– Theme Styles

Available in newer themes, such as Universal Theme - 42 and Mobile - 51, 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.

– Theme Roller

Adjust a number of style attributes and see changes applied to your application in real time. Once you are satisfied with the result, you can save your changes as a Theme Style directly to your application.

See "Using Theme Roller" on page 13-4.

– Template Options

Easily customize how buttons, regions, lists, and other template-based components appear. Using template options, you can displays a single template in several different ways using fewer templates but building far more versatile applications.

See "Understanding Template Options" on page 13-6.

– Theme Subscriptions

See "Understanding Theme Subscriptions" on page 13-13.

■ Interactive Report Enhancements

Easily create multiple interactive reports on a single page. Interactive reports feature a new user interface for displaying your data, report settings, support for fixed column headers, and is more accessible than ever before.

See "Developing Reports" on page 10-1.

■ Modal Dialog

You can easily define modal (and non-modal) pages complete with the ability to utilize standard page processes.

■ Better Navigation

Static or dynamic lists can be used as the basis for your application's navigation menu and navigation bar. Navigation Menus can be positioned to the top or side of your pages, and can be rendered based on any List template, enabling nested navigation items, keyboard support, and greater accessibility.

New Calendar

A calendar component comes with built-in support for month, week, day, and agenda views and is much easier to customize. The calendar is based on the popular FullCalendar library and supports drag and drop, time-based events, and is even responsive.

See "Creating Calendars" on page 12-1.

Mobile Reporting

Build reports that display all of your data on any mobile device, by using reflow table or column toggle. Reflow table wraps each column or changes to displaying multiple lines on very small screens. Column toggle enables you to specify the most important columns, and those which will be hidden as necessary on smaller screens.

See "Developing Reports" on page 10-1.

Static File Upload

Files uploaded into Shared Components, plug-ins, or themes can be cached by your browser and referenced using relative file URLs. This also enables you to upload a ZIP file. The contents will be automatically extracted and relative file references will stay intact. See "Managing Static Application Files" on page 18-8 and "Managing Static Application Files" on page 18-8.

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 release 5.0 functionality.

Oracle Application Express Application Builder User’s Guide features a new organizational structure that explains how to edit pages, components, and controls in both Page Designer and Component View.

See "Understanding Application Pages" on page 2-4, "Managing Pages in Page Designer" on page 8-1, and "Managing Pages in Component View" on page 9-1.
This section offers a quick introduction to using and logging in to Oracle Application Express. The topics in this section assume you have completed the installation process.

See Also: Oracle Application Express Installation Guide

- Understanding Oracle Application Express
- How Oracle Application Express Works
- About Application Express User Roles
- Signing In To Oracle Application Express
- Understanding the Workspace Home Page
- Editing Your Account Profile
- Editing User Preferences
- Using Oracle Application Express Documentation

Understanding Oracle Application Express

This section describes Oracle Application Express and explains the differences between runtime and full development environments.

- What is Oracle Application Express?
- How Oracle Application Express Works
- About the Differences Between Runtime and Full Development Environments

What is Oracle Application Express?

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

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

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

**How Oracle Application Express Works**

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

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

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

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

**About the Differences Between Runtime and Full Development Environments**

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

- **Runtime environment.** In a runtime environment users can run applications but cannot modify them. 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 Application Builder environment described in this document.

**See Also:** "About the Oracle Application Express Runtime Environment" in *Oracle Application Express Installation Guide* and "Installing Exported Applications into a Runtime Environment" in *Oracle Application Express Administration Guide*

**Signing In To Oracle Application Express**

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

This section describes how to sign in to Oracle Application Express. In addition to describing the sign in steps, this section also discusses browser requirements, available user roles, and configuration option.
Browser Requirements

Because Oracle Application Express relies upon standards-compliant HTML5, CSS3, and JavaScript, Oracle recommends that you use the latest web browser software available for the best experience. To learn more, see "Browser Requirements" in Oracle Application Express Installation Guide.

About Application Express User Roles

To access the Oracle Application Express development environment, users sign in to a shared work area called a workspace. Users are divided into four primary roles:

- **End users** do not have access to development or administrative capabilities. End users cannot sign into a workspace and create applications. End users can only run existing database or Websheet applications.

- **Developers** are users who create and edit applications.

- **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. Instance administrators manage workspace provisioning, configure features and instance settings, and manage security.

  **Tip:** The role of Instance administrator is not available in Oracle Database Cloud Service (Database Schema).

  **See Also:** "Managing Users in a Workspace" and "Managing Users Across an Application Express Instance" Oracle Application Express Administration Guide

Overview of Configuring Oracle Application Express

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

How you set up Oracle Application Express depends upon your user role. If you are a developer accessing a hosted development environment, an administrator must grant you access to a workspace. If you are an Instance administrator, you must sign in to the Oracle Application Express Administration Services application and perform the steps described in this section.

- **Signing In To Oracle Application Express Administration Services**

- **Specifying a Provisioning Mode**
1. In a web browser, navigate to the Oracle Application Express Administration Services application. By default, Oracle Application Express Administration Services installs to the following location:

   ■ If your setup uses Oracle REST Data Services, go to:
     
     http://hostname:port/apex/apex_admin

     Where:
     
     - `hostname` is the name of the system where Oracle REST Data Services is installed.
     - `port` is the port number assigned when configuring Oracle REST Data Services. In a default installation, this number is 8080. To learn more, see Oracle REST Data Services Installation, Configuration, and Development Guide.
     - `apex` is the service name defined when configuring Oracle REST Data Services.

   ■ If your setup uses the embedded PL/SQL gateway, go to:
     
     http://hostname:port/apex/apex_admin

     Where:
     
     - `hostname` is the name of the system where Oracle XML DB HTTP Server is installed.
     - `port` is the port number assigned to HTTP on the Oracle XML DB HTTP Server. In a default installation, this number is 8080. If you are using the Oracle Database 12c Pluggable Databases architecture, then each PDB will have a distinct port number. See “Verifying the Port Assigned to HTTP on the Oracle XML DB HTTP Server” on page 1-9.
     - `apex` is the Database Access Descriptor (DAD) defined in the configuration file.

   ■ If your setup uses Apache and `mod_plsql`, go to:
     
     http://hostname:port/pls/apex/apex_admin

     Where:
     
     - `hostname` is the name of the system where Oracle HTTP Server is installed.
     - `port` is the port number assigned to Oracle HTTP Server. In a default installation, this number is 7777.
     - `pls` is the indicator to use the `mod_plsql` cartridge.
– apex is the Database Access Descriptor (DAD) defined in the mod_plsql configuration file.

For users who have upgraded from earlier releases, or who have a custom configuration, this value may be htmldb or something else. Verify your DAD with your Instance administrator.

The Login page appears.

2. In Username, enter the username of an Instance administrator.

3. In Password, enter your Instance administrator account password.

   **Tip:** admin is the default Instance administrator account. As a post installation step, administrators can run the script apxchpwd.sql to create one or more Instance administrator accounts or change passwords for these accounts. To learn more, see the "Creating or Updating Your Instance Administration Account” section for your Web Listener in Oracle Application Express Installation Guide. To create and manage additional Instance administrator accounts, see also "Creating New User Accounts in Administration Services” in Oracle Application Express Administration Guide.

4. Click **Sign In to Administration**.

   Oracle Application Express Administration Services appears.

   **See Also:** "Creating New User Accounts" in Oracle Application Express Administration Guide

**Specifying a Provisioning Mode**

The Instance administrator determines how the process of provisioning (or creating) a workspace works for a specific Oracle Application Express instance. To determine how provisioning works, an Instance Administrator selects one of the following options on the Instance Settings page:

- **Manual** - In this mode, an Instance administrator creates new workspaces and notifies the Workspace administrator of the login icon.

- **Request** - Users request workspaces directly in a self-service fashion. Users click a link on the login page to access a request form. After the workspace request has been granted, users are automatically emailed the appropriate login information.

- **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. Clicking this link validates the user’s email address and then the request is processed. Then another email is sent to the user containing login credentials (that is, the workspace name, username, and password).

   **See Also:** "Creating Workspaces” in Oracle Application Express Administration Guide

**Creating a Workspace**

A workspace enables multiple users to work within the same Oracle Application Express installation while keeping their objects, data and applications private. Each workspace has a unique ID and name. An Oracle Application Express administrator can create a workspace manually or have users submit requests.
Signing In To Oracle Application Express

See Also: "Creating Workspaces" and "Managing Workspace and Change Requests" in Oracle Application Express Administration Guide

Signing In To the Workspace
Once a workspace has been created in Oracle Application Express Administration Services, the final step is to sign in to your workspace to the Oracle Application Express Login page and sign in to that workspace.

See Also: "Signing In To a Workspace" on page 1-7

Signing In and Out of a Workspace as a Developer

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

When you sign in to Oracle Application Express, you sign in to a workspace. If you are a developer, an administrator must grant you access to a workspace.

- Requesting a Workspace
- Signing In To a Workspace
- Signing Out of a Workspace
- Recovering Your Workspace Name
- Resetting Your Password from the Sign In Page

Note: Before users can request a workspace or change their passwords, an Oracle Application Express administrator must configure Oracle Application Express environment preferences.

See Also: "Managing Environment Settings" in Oracle Application Express Administration Guide and "Changing Your Password" on page 1-18

Requesting a Workspace

Note: This section applies only if your Oracle Application Express administrator has configured Oracle Application Express to support workspace requests.

See Also: "Creating Workspaces" in Oracle Application Express Administration Guide

Before you can sign in to Oracle Application Express, an administrator must grant you access to a workspace. Each workspace has a unique ID and name.

To request a workspace:

1. In a web browser, navigate to the Oracle Application Express Sign In page.
Tip: If you do not know the URL for Oracle Application Express Login page, see step 1 in "Signing In To a Workspace" on page 1-7.

The Sign In page appears.

2. Under Workspace, click Request a Workspace.
   The Request Service Wizard appears.

3. Select the type of workspace. Options include:
   - Application Development
   - Packaged Apps Only

4. Follow the on-screen instructions.

See Also: "Understanding the Workspace Home Page" on page 1-10, "Using a Packaged Apps Only Workspace" on page 3-26, and "Creating Workspaces" in Oracle Application Express Administration Guide

Signing In To a Workspace

After an Oracle Application Express administrator approves a workspace request, an e-mail arrives with your login credentials (the workspace name, user name, and password).

Note that if your administrator selected Email Verification as the automated method for handling new workspace requests, you might first receive an email containing a verification link. This step ensures that your email is a valid one before the workspace request is approved.

See Also: "Creating Workspaces" in Oracle Application Express Administration Guide

To sign in to Oracle Application Express:

1. In a web browser, navigate to the Oracle Application Express Login page.
   - If your setup uses the Oracle REST Data Services, go to:
     http://hostname:port/apex

     Where:
     - hostname is the name of the system where Oracle REST Data Services is installed.
     - port is the port number assigned to Oracle REST Data Services. In a default installation, this number is 8080. To learn more, see Oracle REST Data Services Installation, Configuration, and Development Guide
     - apex is the service name defined when configuring the Oracle REST Data Services.
   - If your setup uses the embedded PL/SQL gateway, go to:
     http://hostname:port/apex

     Where:
     - hostname is the name of the system where Oracle XML DB HTTP Server is installed.
Signing In To Oracle Application Express

- **port** is the port number assigned to HTTP on the Oracle XML DB HTTP Server. In a default installation, this number is 8080. See "Verifying the Port Assigned to HTTP on the Oracle XML DB HTTP Server" on page 1-9.

- **apex** is the Database Access Descriptor (DAD) defined in the configuration file.

  For users who have upgraded from earlier releases, or who have a custom configuration, this value may be `htmldb` or something else. Verify your DAD with your Oracle Application Express administrator.

  - If your setup uses Apache and `mod_plsql`, go to:
    
    \[
    \text{http://hostname:port/pls/apex}
    \]

    Where:
    
    - **hostname** is the name of the system where Oracle HTTP Server is installed.
    
    - **port** is the port number assigned to Oracle HTTP Server. In a default installation, this number is 7777. You can find information about your Oracle HTTP Server installation’s port number from either of the following files:
      
      \[
      \text{ORACLE_BASE\ORACLE_HOME\install\portlist.ini}
      \]
      
      \[
      \text{ORACLE_BASE\ORACLE_HTTPSERVER_HOME\Apache\Apache\conf\httpd.conf}
      \]

    Be aware that if you change a port number, it is not updated in the `portlist.ini` file. You can only rely on this file immediately after installation.

    - **pls** is the indicator to use the `mod_plsql` cartridge.

    - **apex** is the Database Access Descriptor (DAD) defined in the `mod_plsql` configuration file.

      For users who have upgraded from earlier releases, or who have a custom configuration, this value may be `htmldb` or something else. Verify your DAD with your Oracle Application Express administrator.

      **See Also:** "Managing Oracle Database Port Numbers" in Oracle Database Installation Guide and `ORACLE_BASE\ORACLE_HTTPSERVER_HOME\Apache\modplsql\conf\dads.readme` for more information on Database Access Descriptor

2. On the Sign In page:

   - In the Workspace field, enter the name of your workspace.
   
   - In the Username field, enter your user name.
   
   - In the Password field, enter your case-sensitive password.

3. Click **Sign In**.

   The workspace home page appears.

   **Tip:** Depending on your setup, you might be required to change your password when you sign in for the first time.

   **See Also:** "Understanding the Workspace Home Page" on page 1-10 and "Changing Your Password" on page 1-18
Verifying the Port Assigned to HTTP on the Oracle XML DB HTTP Server

The embedded PL/SQL gateway runs in the Oracle XML DB Protocol Server in the Oracle database. To verify the port number assigned to HTTP on the Oracle XML DB HTTP Server:

1. Start SQL*Plus and connect to the database where Oracle Application Express is installed as SYS specifying the SYSDBA role. For example:
   - On Windows:
     ```sql
     SYSTEM_DRIVE:\ sqlplus /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```
   - On UNIX and Linux:
     ```sql
     $ sqlplus /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

2. Enter the following statement to verify the port number:
   ```sql
   SELECT DBMS_XDB.GETHTTPPORT FROM DUAL;
   ```
   If the port number returns 0, the Oracle XML DB HTTP Server is disabled.

See Also: Oracle Application Express Installation Guide to learn how to enable the Oracle XML DB HTTP Server

Signing Out of a Workspace

To sign out of Oracle Application Express workspace:

- Select the Account Menu in the upper right corner of the window and click Sign out.

See Also: "Account Menu" on page 1-16

Recovering Your Workspace Name

If you cannot remember your workspace name, you can request a list of all workspace names associated with your email address.

To find your workspace name:

1. On the Sign In page, click Reset Password.
2. Click Find My Workspace.
3. Enter your email address and click Find Workspace.
   You will receive an email listing all workspaces associated with the email address you provided.

Resetting Your Password from the Sign In Page

To reset your password from the Sign In page, you must provide your email address and workspace name.

To reset your password from the Sign In Page:

1. Under Sign in, click Reset Password.
2. Enter your email address and workspace name, and click Reset Password.
You will receive an email containing new passwords for all accounts associated with the email address and workspace name you provided.

3. On the Sign In page, enter your workspace, username, and the new password and click Sign In.
   
   **Tip:** Passwords are case sensitive. Do not copy and paste the new password.

4. On Change Password, enter:
   a. Current Password - Enter the current password provided by the email.
   b. New Password - Enter your new password.
   c. Confirm Password - Enter your new password again.
   d. Click Apply Changes.

   **See Also:** "Changing Your Password" on page 1-18

**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
- News and Messages
- Top Applications and Top Users Dashboards
- Developer Navigation Tools
- Available Updates Region
- Accessibility Mode Region
The following large icons display in the center of the page:

- **Application Builder.** Use Application Builder to create an application, composed of a set of HTML pages, based on database objects. See "Application Builder Concepts" on page 2-1 and "Understanding Application Builder" on page 4-1.

- **SQL Workshop.** Use the SQL Workshop to access tools for viewing and managing database objects. See Oracle Application Express SQL Workshop Guide.

- **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. See "Managing the Application Life Cycle with Team Development" on page 5-1.

- **Packaged Apps.** Links to the Packaged Apps page. See "Utilizing Packaged Applications" on page 3-1.

An menu bar also displays at the top of the page. Click a menu to access an alternate navigation path to different components within Application Builder, SQL Workshop, Team Development, and Packaged Apps.
Top Applications and Top Users Dashboards

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

News and Messages

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 (>).

Both workspace announcements and system messages include a gold icon to the left of message text. A textual identifier, either System Message or Workspace Announcement, also displays beneath the message text.

See Also: "Managing News Entries" on page 5-22 and "Defining a System Message" and "Defining a Workspace Announcement" in Oracle Application Express Administration Guide

Developer Navigation Tools

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

- Global Search Field
- Administration Menu
- Help Menu
- Account Menu

Global Search Field

The Search field displays in the header region in upper right corner of most Application Builder pages. By default, this Search field supports regular expressions. How the search field works depends upon the page you are on. Consider the following examples:

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

Within Application Builder the following special syntax can be used 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
```

**Administration Menu**

![Administration Menu](image)

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. See "Workspace and Application Administration" in Oracle Application Express Administration Guide.
- **Manage Service** displays a submenu of options on the Manage Service page. See "Manage Service Page" in Oracle Application Express Administration Guide.
- **Manage Users and Groups** link to Manage Users and Groups page. See "Managing Users in a Workspace" in Oracle Application Express Administration Guide
- **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. See "Monitoring Activity Within a Workspace" in Oracle Application Express Administration Guide.

- **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. See "Viewing Workspace Dashboards" in Oracle Application Express Administration Guide.

- **Change My Password** - Links to the Change Password section of the Edit Profile dialog box. See "Changing Your Password" on page 1-18.

### Help Menu

The Help menu resembles a question mark. Clicking Help displays the following menu items:

- **Documentation.** Links to Oracle Application Express, Release 5.0 Oracle Help Center. See "Using Oracle Application Express Documentation" on page 1-19.

- **Discussion Forum.** Link to Application Express Discussion Forum:
  
  https://community.oracle.com/community/database/developer-tools/application_express

- **Oracle Technology Network.** Link to Oracle Application Express page on Oracle Technology Network:
  

- **About.** Links links to the About Application Express dialog. This dialog includes information about the Oracle Application Express instance and the database version.
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. See "Editing Your Account Profile and Preferences" on page 1-17.

  **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 Application Builder runs applications. See "Editing User Preferences" on page 1-18.

- **Sign Out.** Click Sign Out to log out the current workspace. See "Signing Out of a Workspace" on page 1-9.

**See Also:** “Editing Your Account Profile” on page 1-17

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:

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: "Accessibility in Oracle Application Express" on page E-1, "Understanding Screen Reader Mode" on page E-1, and "Understanding High Contrast Mode" on page E-4

Editing Your Account Profile

The section describes how to edit your account profile. By editing your account profile you can update your email address, edit your first or last name, upload a photo, or change your password.

- Editing Your Account Profile and Preferences
- Changing Your Password

See Also: "Resetting Your Password from the Sign In Page" on page 1-9

Editing Your Account Profile and Preferences

To edit your account profile:

1. Sign in to Oracle Application Express.
2. Click the Account menu in the header region. See "Account Menu" on page 1-16.
3. Click Edit Profile.

The Edit Profile dialog appears.

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

   To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

5. Under Profile Photo, add, change, or remove your photo.

7. Click Apply Changes.

Changing Your Password

To reset your password:

1. Sign in to Oracle Application Express.

2. Click the Account menu in the header region and select Edit Profile.

   The Edit Profile dialog appears.

3. Scroll down to Change Password and edit the following:
   - Enter Current Password
   - Enter New Password
   - Confirm Password

4. Click Apply Changes.

   See Also: "Changing an End User Password" in Oracle Application Express Administration Guide and "Account Menu" on page 1-16

Editing User Preferences

The section describes how to edit your user preferences. By editing your user preferences, you can update the default workspace schema, specify a default date format, and determine how Oracle Application Express manages windows and tabs when running applications.

To edit your user preferences:

1. Sign in to Oracle Application Express.

2. Click the Account menu in the header region and click Preferences.

   The Preferences dialog appears.

3. Edit the following attributes:
   - Default Schema - Specifies the default schema used for data browsing, application creation, and SQL script execution. When using workspaces that have more than one schema available, this schema is the default for creating
applications, performing SQL queries, and so on. This feature does not control security, only the user's preference.

- Default Date Format - Specify the default Oracle date format for the user. This will control the default date format within SQL Workshop.

- Run application in new window - Specify how Oracle Application Express manages windows or tabs when running applications from the Application 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 Application 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.

4. Click **Apply Changes**.

---

**Using Oracle Application Express Documentation**

Oracle Application Express provides a hosted online documentation library and Field-Level Help. This section describes how to access and search the library, use the library master index, download the documentation library, and use field-level Help.

- Accessing Oracle Help Center
- Searching the Oracle Help Center
- Downloading Online Documents
- Searching a Downloaded Documentation Library
- Viewing Field-Level Help

**Accessing Oracle Help Center**

To access the online Oracle Application Express documentation library:

1. Click the **Help** menu in the upper right corner.
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 click Search.
      Search results display.
   b. Select a topic to review.

4. To search a specific document:
   a. Select a document.
   b. Enter keywords in the Search field in the upper left corner.
      Search results display.
   c. Select a topic to review.

Downloading Online Documents

You can download the Oracle Application Express documentation library locally from the Oracle Application Express Documentation page.

To download the online documentation library:

1. Go to the Oracle Application Express Documentation page:
2. Locate the Oracle Application Express library you want to download and click the ZIP link.
3. Download ZIP file and extract it locally.
4. Open the extracted folder and click index.html.
   The Oracle Application Express Documentation Library welcome page appears.

   **See Also:**  "Accessing Oracle Help Center" on page 1-19

**Searching a Downloaded Documentation Library**

You can use the provided Acrobat search index to fast search across PDFs in the downloaded library.

   **Tip:**  You must have access to the Internet to be able to use the global search capability and master index as described in "Searching the Oracle Help Center" on page 1-20.

To perform a global search on the downloaded documentation:
1. Go to the location of the downloaded documentation library.
2. Double click index.pdx.
3. Click Shift+Ctrl+F.
   The Search window appears.
4. Enter your search criteria and click Search.
   Search results appear.

**Viewing Field-Level Help**

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

- **Viewing Field-level Help in Oracle Application Express**
- **Viewing Help in Page Designer**

   **See Also:**  "Managing Pages in Page Designer" on page 8-1 and "Managing Pages in Component View" on page 9-1

**Viewing Field-level Help in Oracle Application Express**

To view field-level help:
1. Navigate the attribute you need information about.
   In most of Oracle Application Express, attributes having field-level Help have a light gray Help Text icon that resembles a question mark (?).
2. To view field-level Help, click the Help Text icon.

A Help window appears.

**Tip:** Click the **Help** menu in the header region and select **Documentation** to display the online documentation library. To access field-level help with the keyboard, focus on the correct field and then press **Alt+F1**.

**Viewing Help in Page Designer**

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

To view Help in Page Designer:


   Page Designer appears.

2. In either the Rendering tree or Grid Layout, select a component or control.

   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. 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:** Clicking the Help menu in the header region and selecting Documentation displays the online documentation library.
This section provides basic conceptual information about Application Builder.

- Understanding Oracle Application Express Applications
- Understanding Application Pages
- Running an Application or Page
- How Does Page Processing and Page Rendering Work?
- Understanding Session State Management
- Understanding URL Syntax
- About Enabling Support for Bookmarks
- Managing Session State Values
- Understanding Substitution Strings

See Also: "Understanding Application Builder" on page 4-1

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?
- What is the Difference Between a Database and Websheet Application?
- About Running the Create Application Wizard

See Also: "Understanding Application Builder" on page 4-1

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

An Oracle Application Express application enables you to manage and display data from an Oracle database. You build an application using Application Builder. Using Application 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
- About Websheet Applications

About Database Applications

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

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

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

See Also: "Creating Database Applications" on page 7-1

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" on page 6-1

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 Application Builder home page.

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" on page 7-1 and "Creating a Database Application" on page 7-4.

- **Mobile.** Mobile database applications are designed to run specifically on smartphone devices. See "Using the Create Application Wizard" on page 7-1 and "Creating Mobile Applications" on page 7-10.

- **Websheet.** Websheet applications enable users to build data centric applications without any SQL programming knowledge. Websheet applications support pages, data grids, and reports and support community contributions. See "Creating Websheet Applications" on page 6-1.

- **Packaged Application.** Links to the Packaged Apps gallery. See "About Packaged Applications" on page 3-1.

- **Additional Create Application Options:**
  - **From Spreadsheet.** Upload spreadsheet information as a file or by copying and pasting content. See "Creating a Desktop Database Application from a Spreadsheet" on page 7-7.
  - **Copy of existing Application.** See "Copying a Database Application" on page 7-8.

See Also: "Creating a Database Application" on page 7-4
Understanding Application Pages

A page is the basic building block of an application. This section explains what a page is and how to edit it in two views: Page Designer and Component View.

- What Is an Application Page?
- Understanding the Difference Between Page Designer and Component View
- Viewing a Page in Page Designer
- Viewing a Page in Component View
- Switching from Page Designer to Component View
- Switching from Component View to Page Designer

What Is an Application Page?

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 can view and edit a page in two views: Page Designer and Component View.

See Also: "Managing Pages in Page Designer" on page 8-1 and "Managing Pages in Component View" on page 9-1

Understanding the Difference Between Page Designer and Component View

This section describes key differences between using Page Designer and Component View to edit application pages.

- Page Designer
- Component View
Page Designer

Page Designer is a full featured Integrated Development Environment (IDE) designed to maximize developer productivity in maintaining application pages. The Page Designer window is divided into three main sections.

- **Tree Pane** - The Tree pane displays in the left column and features the following tree views: Rendering, Dynamic Actions, Processing, and Shared Components.

- **Grid Layout** - Grid Layout displays in the center column. Grid Layout is a visual representation of the regions, items, and buttons that define a page. You can edit the position of page components by dragging and dropping them. You can also add new regions, items, and button by dragging and dropping them from Gallery at the bottom of the tab.

- **Property Editor** - The Property Editor displays in the right column. Use the Property Editor to edit the properties of the component currently selected in the Tree Pane or Grid Layout.

**Tip:** The Rendering tree and Grid Layout are two different representations of the regions, items, and buttons that define a page.

**See Also:** "Managing Pages in Page Designer" on page 8-1
Component View

Component View is the traditional user interface for editing a page in Application Builder. 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.

The 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: "Managing Pages in Component View" on page 9-1

Viewing a Page in Page Designer

To view a page in Page Designer:

1. Navigate to the Workspace home page.
2. Click the Application Builder icon.
3. Select an application.
4. Select a page.

The page appears. By default, 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 Component View to Page Designer" on page 2-8, "Running an Application or Page" on page 2-9, and "Managing Pages in Page Designer" on page 8-1

---

**Viewing a Page in Component View**

To view a page in Component View:

1. Navigate to the Workspace home page.
2. Click the **Application Builder** icon.
3. Select an application.
4. Select a page.
   
The default view, Page Designer, appears.
5. Change to Component View:
   
   a. Locate the Page Designer toolbar at the top of the page.
   
   b. Click **Component View**.

Component View appears.

You preview a page by running it. Running an application page displays the rendered version as end users would see it.

**See Also:** "Running an Application or Page" on page 2-9, "Switching from Page Designer to Component View" on page 2-7, and "Managing Pages in Component View" on page 9-1

---

**Switching from Page Designer to Component View**

To switch from Page Designer to Component View:

2. Change to Component View:
   
   a. Locate the Page Designer toolbar at the top of the page.
   
   b. Click **Component View** icon.

Component View appears.

**See Also:** "Managing Pages in Component View" on page 9-1
Switching from Component View to Page Designer

To switch from Component View to Page Designer:

1. View the page in Component View. See "Viewing a Page in Component View" on page 2-7
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: "Managing Pages in Page Designer" on page 8-1

Managing User Interface Design

This section describes key concepts about selecting a user interface for your application.

■ About Determining Your Application User Interface
■ About Mobile Application Design
■ About Mobile Application Design

About Determining Your Application User Interface

Application 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" on page 7-4, "Creating Mobile Applications" on page 7-10, and "Creating a Global Page to Display Components on Every Page" on page 14-3

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" on page 7-10 and "About Responsive Design" on page 13-2

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 grid 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" on page 7-10 and "About Responsive Design" on page 13-2

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
- How Your Browser Impacts the Way Applications Run
- Running an Application from the Application Builder Home Page
- Running an Application from the Application Home Page
- Running a Page on the Application Home Page
- Running a Page in Page Designer
- Running a Page in Component View
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 Application Builder. Clicking the **Run Page** button runs the current page or the last selected page.

How Your Browser Impacts the Way Applications Run

When you run an application, the application displays in a new window. Whether that new window is a new tab or new browser windows depends upon how you have configured your web browser. If you are using Google Chrome or Apple Safari, the running application displays in a new tab by default. If you are using Microsoft Internet Explorer or Mozilla Firefox, the application displays in a new browser window by default. You can further control how Oracle Application Express manages tabs and windows when running an application by editing your account preferences.

See Also:  "Editing User Preferences" on page 1-18

Running an Application from the Application Builder Home Page

To run an application from the Application Builder home page:

1. On the Workspace home page, click the **Application 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.
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 Application Builder icon.
   The Application 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.

Running a Page on the Application Home Page

To run a page from the Pages list:

1. On the Workspace home page, click the Application 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.

Running a Page in Page Designer

To run a page in Page Designer, click the Save and Run Page button in the upper right corner. The Save and Run Page button resembles a blue right-facing arrow.

Running a Page in Component View

To run a page in Component View, click the Run Page button in the upper right corner.

How Does Page Processing and Page Rendering Work?

When you create an application in Application 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
- Understanding Conditional Rendering and Processing
- About Verifying User Identity
- About Controlling Access to Controls and Components

How the Application Express Engine Renders and Processes Pages

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

- **Show Page** is the page rendering process. It assembles all the page attributes (including regions, items, and buttons) into a viewable HTML page.

- **Accept Page** performs page processing. It performs any computations, validations, processes, and branching.

When you request a page using a URL, the engine is running **Show Page**. 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.

Understanding Conditional Rendering and Processing

This section describes how you can use conditions in your application to control how pages and page components display and when processes, computations, and validations execute.

- What is a Condition?
- Selecting a Condition
- Condition Example: Current Page in Expression 1
- Condition Example: Exists
- Condition Example: PL/SQL Expression

See Also:

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" on page C-1 for a detailed listing of available condition types
Selecting a Condition
You specify a condition by selecting a condition type. You can select a condition type when you first create the control or component, or later when editing the control or component.

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

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

3,100,203

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

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

SELECT 1 FROM employees WHERE department_id = :P101_DEPTNO

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

See Also:  "Referencing Session State Using Bind Variable Syntax" on page 2-24

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

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

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

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

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

See Also:  "Establishing User Identity Through Authentication" on page 20-37
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" on page 20-49

Understanding Session State Management

This section describes session statement management in Oracle Application Express.

- What is Session State?
- What Is a Session?
- About Session IDs
- About Referencing Session State

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.

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
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:


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"

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
- Using the Session Page

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" on page 2-15

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 describes the information that displays on the Session page depending upon the View option you select.

<table>
<thead>
<tr>
<th>View Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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" on page 15-1                                                                                     |
| 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" on page 18-15 and "Understanding Substitution Strings" on page 2-25 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: "Viewing Session State for a Page" on page 2-15

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
- About Using f?p Syntax to Link Pages
- Application Alias
- Page Alias
Understanding URL Syntax

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:


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" on page 2-14 and "About Publishing the Database Application URL" on page 24-30

About Using f?p Syntax to Link Pages

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


Table 2–2 describes the arguments you can pass when using f?p syntax.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>Indicates an application ID or alphanumeric application alias.</td>
</tr>
<tr>
<td>Page</td>
<td>Indicates a page number or alphanumeric page alias.</td>
</tr>
<tr>
<td>Session</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>Short substitution string: &amp;SESSION.</td>
</tr>
<tr>
<td></td>
<td>PL/SQL: V('SESSION')</td>
</tr>
<tr>
<td></td>
<td>Bind variable: :APP_SESSION</td>
</tr>
</tbody>
</table>

See Also: "Understanding Session State Management" on page 2-14 and "About Publishing the Database Application URL" on page 24-30
### Understanding URL Syntax

#### Request
Sets the value of `REQUEST`. Each application button sets the value of `REQUEST` to the name of the button which enables accept processing to reference the name of the button when a user clicks it. You can reference `REQUEST` using the syntax:

- Substitution string: `&REQUEST`.
- PL/SQL: `V('REQUEST')`
- Bind variable: `:REQUEST`

#### Debug
Displays application processing details. Valid values for the DEBUG flag include:

- YES.
- LEVELn
- NO

Setting this flag to YES displays details about application processing. Setting this flag to LEVELn (where n is between 1 and 9) controls the level of debug detail that displays. The value of YES equals LEVEL4.

You can reference the Debug flag using the following syntax:

- Short substitution string: `&DEBUG`.
- PL/SQL: `V('DEBUG')`
- Bind variable: `:DEBUG`

See Also: "Debugging an Application" on page 22-1

#### ClearCache
Clears the cache. This sets the value of items to null.

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.

See Also: "Clearing Session State" on page 2-22

#### itemNames
Comma-delimited list of item names used to set session state with a URL.

#### itemValues
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:

```
\123,45\`
```

Every character sequence except backslash comma (`,`) can be enclosed with backslash.

---

**Table 2–2 (Cont.) f?p Syntax Arguments**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>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:</td>
</tr>
</tbody>
</table>
|          | - Substitution string: `&REQUEST`.
|          | - PL/SQL: `V('REQUEST')`
|          | - Bind variable: `:REQUEST` |
| Debug    | Displays application processing details. Valid values for the DEBUG flag include: |
|          | - YES. |
|          | - LEVELn |
|          | - NO |
|          | Setting this flag to YES displays details about application processing. Setting this flag to LEVELn (where n is between 1 and 9) controls the level of debug detail that displays. The value of YES equals LEVEL4. |
|          | You can reference the Debug flag using the following syntax: |
|          | - Short substitution string: `&DEBUG`.
|          | - PL/SQL: `V('DEBUG')`
|          | - Bind variable: `:DEBUG` |

See Also: "Debugging an Application" on page 22-1

---

**2-18 Oracle Application Express Application Builder User's Guide**
It is important to understand how f?p syntax works. Application 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.

### 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" on page 7-13 and "Calling a Page Using an Application and Page Alias" on page 2-19

### 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:** "Editing Page Attributes in Page Designer" on page 8-19, "Editing Page Attributes in Component View" on page 9-11, and "Calling a Page Using an Application and Page Alias" on page 2-19

### 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
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" on page 15-42 and "APP_SESSION" on page 2-32

About Enabling Support for Bookmarks

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, release 5.0 includes a new feature called Rejoin Sessions that enables you to control this type of support at the page, application, or instance-level. To learn more, see "About Rejoin Sessions" on page 20-5.

See Also: "Facilitating Bookmarks by Using Zero as the Session ID" in Oracle Application Express Application Builder User’s Guide, release 4.2

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
- Setting Session State
- Clearing Session State
- Referencing Session State Using Bind Variable Syntax

See Also: "Referencing Item Values" on page 15-2

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 describes the supported syntax for referencing item values.
Managing Session State Values

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**
- **About Setting Session State with a Form Submission**

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" on page 2-21.
- Computation. See "Understanding Application Computations" on page 17-30.

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

---

**Table 2–3 Syntax for Referencing Item Values**

<table>
<thead>
<tr>
<th>Type</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL</td>
<td>Standard item syntax: <code>:MY_ITEM</code></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Syntax for items containing special characters: <code>:&quot;MY_ITEM&quot;</code></td>
<td></td>
</tr>
<tr>
<td>PL/SQL</td>
<td><code>V('MY_ITEM')</code></td>
<td>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.</td>
</tr>
<tr>
<td>Static text (exact)</td>
<td>Standard item syntax: <code>&amp;MY_ITEM</code></td>
<td>For static text or an exact substitution, use the convention &amp;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.</td>
</tr>
<tr>
<td></td>
<td>Syntax for items containing special characters: <code>&amp;&quot;MY_ITEM&quot;</code></td>
<td></td>
</tr>
</tbody>
</table>

See Also: Oracle Application Express API Reference

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: Oracle Application Express API Reference

---
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, "Syntax for Referencing Item Values" on page 2-21.

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
- Clearing Cache for All Page Items
- Example: Clearing Cache for an Entire Application
- About Resetting an Application Completely
- About Clearing Cache for the Current User Session

About Clearing Cache for an Item

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

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

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
- Example: Clearing Cache on a Page and Passing an Item Value
- Example: Clearing Session Cache on a Page and Passing Values to Multiple Items

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:
Managing Session State Values

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

**Example: 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).

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

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

**Example: 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.

```plaintext
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

**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`.

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

**About Resetting an Application Completely**

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

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

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

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

Referencing Session State Using Bind Variable Syntax

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

- **About Using Bind Variable Syntax**
- **About Using Bind Variables in Regions Based on a SQL Query or LOV**
- **About Using Bind Variables in Regions Based on PL/SQL**

**About Using Bind Variable Syntax**

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.

**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 (plsql 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" on page 14-4 for information about creating regions
Understanding Substitution Strings

This section describes how to use substitution strings within a page template or region source to replace a character string with another value. As you design your application and enable users to edit items, you use substitution strings to pass information.

- About Using Substitution Strings
- Controlling Output Escaping in Substitution Strings
- About Determining Substitution String Usage within Templates
- Using Built-in Substitution Strings

About Using Substitution Strings

You can use substitution strings in Application 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:

&P101_X.

Notice the required trailing period. When the page is rendered, Application Express engine replaces value the substitution string with the value of item P101_X.
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#

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" on page 13-23.

Using Built-in Substitution Strings

Application Builder supports many built-in substitution strings. You can reference these substitution strings to achieve specific types of functionality. This section describes available built-in substitution strings, explains when to use them, and discusses what supported syntax is currently available.
Tip: Note that bind variable :USER has special meaning within the database. Also, the term Direct PL/SQL refers to PL/SQL that can be used in stored database objects such as procedures and functions.

- APEX$ROW_NUM
- APEX$ROW_SELECTOR
- APEX$ROW_STATUS
- APP_ID
- APP_ALIAS
- APP_AJAX_X01, ... APP_AJAX_X10
- APP_DATE_TIME_FORMAT
- APP_IMAGES
- APP_NLS_DATE_FORMAT
- APP_NLS_TIMESTAMP_FORMAT
- APP_NLS_TIMESTAMP_TZ_FORMAT
- APP_NLS_TIMESTAMP_TZ_FORMAT
- APP_PAGE_ALIAS
- APP_PAGE_ID
- APP_REQUEST_DATA_HASH
- APP_SESSION
- APP_SESSION_VISIBLE
- APP_TITLE
- APP_UNIQUE_PAGE_ID
- APP_USER
- BROWSER_LANGUAGE
- CURRENT_PARENT_TAB_TEXT
- DEBUG
- HOME_LINK
- IMAGE_PREFIX
- LOGIN_URL
- LOGOUT_URL
- PRINTER_FRIENDLY
- PROXY_SERVER
- PUBLIC_URL_PREFIX
- REQUEST
- SCHEMA_OWNER
- SQLERRM
- SYSDATE_YYYYMMDD
Understanding Substitution Strings

- WORKSPACE IMAGES
- WORKSPACE_ID

See Also:  "Substitutions" on page 7-19 and "Establishing User Identity Through Authentication" on page 20-37

APEX$ROW_NUM

APEX$ROW_NUM refers the currently processed row number of a submitted tabular form data. You can use this placeholder in validations, processes, and conditions associated with a tabular form to refer to the row number of the currently processed tabular form row. Table 2–4 describes the supported syntax for referencing APEX$ROW_NUM.

<table>
<thead>
<tr>
<th>Table 2–4  APEX$ROW_NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Type</td>
</tr>
<tr>
<td>Bind variable</td>
</tr>
<tr>
<td>PL/SQL</td>
</tr>
<tr>
<td>Substitution string</td>
</tr>
</tbody>
</table>

APEX$ROW_SELECTOR

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

<table>
<thead>
<tr>
<th>Table 2–5  APEX$ROW_SELECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Type</td>
</tr>
<tr>
<td>Bind variable</td>
</tr>
<tr>
<td>PL/SQL</td>
</tr>
<tr>
<td>Substitution string</td>
</tr>
</tbody>
</table>

APEX$ROW_STATUS

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

<table>
<thead>
<tr>
<th>Table 2–6  APEX$ROW_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Type</td>
</tr>
<tr>
<td>Bind variable</td>
</tr>
<tr>
<td>PL/SQL</td>
</tr>
<tr>
<td>Substitution string</td>
</tr>
</tbody>
</table>

APP_ID

APP_ID identifies the application ID of the currently executing application. Table 2–7 describes the supported syntax for referencing APP_ID.
Understanding Substitution Strings

The following is an example of a substitution string reference:

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

APP_ALIAS

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

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

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

Table 2–8 describes the supported syntax for referencing APP_ALIAS.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_ALIAS</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_ALIAS')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_ALIAS.</td>
</tr>
</tbody>
</table>

The following is an HTML example:

Click me to go to page 1 &lt;a href="f?p=&APP_ALIAS.:1:&APP_SESSION."&gt; of the current application&lt;/a&gt;

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.

Table 2–9 describes the supported syntax for referencing APP_AJAX_Xnn.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>... :APP_AJAX_X01, ... :APP_AJAX_X10</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>... v('APP_AJAX_X01'), ... v('APP_AJAX_X10')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>... &amp;APP_AJAX_X01., ... &amp;APP_AJAX_X10.</td>
</tr>
</tbody>
</table>
Understanding Substitution Strings

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

**APP_DATE_TIME_FORMAT**

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

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td><code>:APP_DATE_TIME_FORMAT</code></td>
</tr>
<tr>
<td>PL/SQL</td>
<td><code>V('APP_DATE_TIME_FORMAT')</code></td>
</tr>
<tr>
<td>Substitution string</td>
<td><code>&amp;APP_DATE_TIME_FORMAT.</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td><code>:APP_DATE_TIME_FORMAT</code></td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>Not available.</td>
</tr>
<tr>
<td>PL/SQL</td>
<td><code>V('APP_DATE_IMAGES')</code></td>
</tr>
<tr>
<td>Substitution string</td>
<td><code>&amp;APP_DATE_IMAGES.</code></td>
</tr>
<tr>
<td>Template substitution</td>
<td><code>#APP_DATE_IMAGES#</code></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td><code>:APP_IMAGES</code></td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>Not available.</td>
</tr>
<tr>
<td>PL/SQL</td>
<td><code>V('APP_IMAGES')</code></td>
</tr>
<tr>
<td>Substitution string</td>
<td><code>&amp;APP_IMAGES</code></td>
</tr>
<tr>
<td>Template substitution</td>
<td><code>#APP_IMAGES#</code></td>
</tr>
</tbody>
</table>

See Also:  "IMAGE_PREFIX" on page 2-37, "WORKSPACE_ IMAGES" on page 2-41, and "About Managing Images" on page 14-25

**APP_NLS_DATE_FORMAT**

APP_NLS_DATE_FORMAT is the application date format of the database session. This value reflects the format specified in the Application Date Format attribute of the Globalization settings of the application. However, if the Application Date Format is not set, then `APP_NLS_DATE_FORMAT` returns the NLS_DATE_FORMAT value of the database session at the start of the request to the Application Express engine. Table 2–12 describes the supported syntax for referencing `APP_NLS_DATE_FORMAT`.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td><code>:APP_NLS_DATE_FORMAT</code></td>
</tr>
<tr>
<td>PL/SQL</td>
<td><code>V('APP_NLS_DATE_FORMAT')</code></td>
</tr>
</tbody>
</table>
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 returns the NLS_TIMESTAMP_FORMAT value of the database session at the start of the request to the Application Express engine. Table 2–13 describes the supported syntax for referencing APP_NLS_TIMESTAMP_FORMAT.

Table 2–13 APP_NLS_TIMESTAMP_FORMAT Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_NLS_TIMESTAMP_FORMAT</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_NLS_TIMESTAMP_FORMAT')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_NLS_TIMESTAMP_FORMAT.</td>
</tr>
</tbody>
</table>

APP_NLS_TIMESTAMP_TZ_FORMAT

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

Table 2–14 APP_NLS_TIMESTAMP_TZ_FORMAT Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_NLS_TIMESTAMP_TZ_FORMAT</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_NLS_TIMESTAMP_TZ_FORMAT')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_NLS_TIMESTAMP_TZ_FORMAT.</td>
</tr>
</tbody>
</table>

APP_PAGE_ALIAS

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

Table 2–15 APP_PAGE_ALIAS Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_PAGE_ALIAS</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>v('APP_PAGE_ALIAS')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_PAGE_ALIAS.</td>
</tr>
</tbody>
</table>
The following is an HTML example:
The alias of the current page is: &APP_PAGE_ALIAS.

APP_PAGE_ID

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

Table 2–16  APP_PAGE_ID Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_PAGE_ID</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>:APP_PAGE_ID</td>
</tr>
<tr>
<td>PL/SQL and Direct PL</td>
<td>NV('APP_PAGE_ID')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_PAGE_ID.</td>
</tr>
</tbody>
</table>

The following is an example of a substitution string reference:
f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.

APP_REQUEST_DATA_HASH

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

Table 2–17  APP_REQUEST_DATA_HASH Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_REQUEST_DATA_HASH</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_REQUEST_DATA_HASH')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_REQUEST_DATA_HASH.</td>
</tr>
</tbody>
</table>

APP_SESSION

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

Table 2–18  APP_SESSION Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_SESSION</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_SESSION')</td>
</tr>
<tr>
<td>Short PL/SQL</td>
<td>V('SESSION')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_SESSION.</td>
</tr>
<tr>
<td>SYS_CONTEXT variable</td>
<td>SYS_CONTEXT('APEX$SESSION', 'APP_SESSION')</td>
</tr>
</tbody>
</table>
Consider the following examples:

- From within an HTML region:
  <a href="f?p=100:5:&APP_SESSION_VISIBLE."">click me</a>

- Using PL/SQL:
  
  \[
  \text{htf.anchor('f?p=100:5:||V('APP_SESSION_VISIBLE'),'click me');}
  \]

- Using a SQL query:
  SELECT htf.anchor('f?p=100:5:||:APP_SESSION_VISIBLE,'click me') FROM DUAL;

- Using the \texttt{SYS_CONTEXT} variable:
  
  \[
  \text{SELECT ... WHERE apex_session_id = SYS_CONTEXT('APEX$SESSION', 'APP_SESSION_ID')}
  \]

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

**APP_SESSION_VISIBLE**

\texttt{APP_SESSION_VISIBLE} is similar to the built-in substitution \texttt{APP_SESSION}. Use this substitution string to create hypertext links between application pages that maintain a session state by passing the session number. \texttt{APP_SESSION_VISIBLE} always returns '0' when users are not authenticated to an application and they are using the Zero Session ID feature of Oracle Application Express. Table 2–19 describes the supported syntax for referencing \texttt{APP_SESSION_VISIBLE}.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_SESSION_VISIBLE</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_SESSION_VISIBLE')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_SESSION_VISIBLE</td>
</tr>
</tbody>
</table>

Consider the following examples:

- From within an HTML region:
  <a href="f?p=100:5:&APP_SESSION_VISIBLE."">click me</a>

- Using PL/SQL:
  
  \[
  \text{sys.htf.anchor('f?p=100:5:||V('APP_SESSION_VISIBLE'),'click me');}
  \]

- Using a SQL query:
  SELECT sys.htf.anchor('f?p=100:5:||:APP_SESSION_VISIBLE,'clickme') FROM DUAL;

**APP_TITLE**

\texttt{APP_TITLE} is an alphanumeric title for the current application. The title is derived from an application substitution string called \texttt{APP_TITLE}. If not defined the Logo attribute will be used if it is of type text. The last fallback is the application name. Table 2–20 describes the supported syntax for referencing \texttt{APP_TITLE}.
The following is an HTML example:

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

**APP_UNIQUE_PAGE_ID**

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

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_UNIQUE_PAGE_ID</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_UNIQUE_PAGE_ID')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_UNIQUE_PAGE_ID.</td>
</tr>
</tbody>
</table>

The following is an HTML example:

SELECT 'f?p=100:1:'||:APP_SESSION||':'||:APP_UNIQUE_PAGE_ID||':P1_EMPNO:'||employee_id,
first_name,
job_id
FROM employees

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

**APP_USER**

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

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:APP_USER</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('APP_USER')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;APP_USER.</td>
</tr>
</tbody>
</table>
Consider the following examples:

- From within an HTML region:

  ```html
  Hello you are logged in as &APP_USER.
  ```

- Using PL/SQL:

  ```sql
  htp.p('Hello you are logged in as'||V('APP_USER'));
  ```

- As a bind variable:

  ```sql
  SELECT * FROM some_table WHERE user_id = :APP_USER
  ```

- Using the `SYS_CONTEXT` variable:

  ```sql
  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" on page 7-20 for information about the Public User attribute

**AUTHENTICATED_URL_PREFIX**

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

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:AUTHENTICATED_URL_PREFIX</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('AUTHENTICATED_URL_PREFIX')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;AUTHENTICATED_URL_PREFIX</td>
</tr>
</tbody>
</table>

**BROWSER_LANGUAGE**

`BROWSER_LANGUAGE` refers to the web browser's current language preference. Table 2–24 describes the supported syntax for referencing `BROWSER_LANGUAGE`.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:BROWSER_LANGUAGE</td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_BROWSER_LANGUAGE</td>
</tr>
</tbody>
</table>
Understanding Substitution Strings

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

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>Not Available.</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;CURRENT_PARENT_TAB_TEXT.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:DEBUG</td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_DEBUG</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('DEBUG')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;DEBUG.</td>
</tr>
</tbody>
</table>

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

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

HOME_LINK
HOME_LINK is the home page of an application. The Application Express engine redirects to this location if no page is given and if no alternative page is dictated by the authentication scheme’s logic. You define the Home Link on the Application Attributes page. Table 2–27 describes the supported syntax for referencing HOME_LINK.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_HOME_LINK</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('HOME_LINK')</td>
</tr>
</tbody>
</table>
The value of `IMAGE_PREFIX` determines the virtual path the web server uses to point to the images directory distributed with Oracle Application Express. To reference uploaded images, use `WORKSPACE_IMAGES` and `APP_IMAGES`. Table 2–28 describes the supported syntax for referencing `IMAGE_PREFIX`.

See Also: "APP_IMAGES" on page 2-30, "WORKSPACE_IMAGES" on page 2-41, and "Editing the Application Attributes" on page 7-14

### Table 2–28  IMAGE_PREFIX Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:IMAGE_PREFIX</td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_IMAGE_PREFIX</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('IMAGE_PREFIX')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;IMAGE_PREFIX.</td>
</tr>
<tr>
<td>Template Substitution</td>
<td>#IMAGE_PREFIX#</td>
</tr>
</tbody>
</table>

LOGOUT_URL

Use LOGOUT_URL to display a link to a logout page for users that are not currently logged in. Table 2–29 describes the supported syntax for LOGOUT_URL.

See Also: "Authentication" on page 7-20 and "Security Attributes Page" on page 7-20

### Table 2–29  LOGIN_URL Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:LOGIN_URL</td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_LOGIN_URL</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('LOGIN_URL')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;LOGIN_URL.</td>
</tr>
<tr>
<td>Template Substitution</td>
<td>#LOGIN_URL#</td>
</tr>
</tbody>
</table>

LOGOUT_URL

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

Table 2–30  LOGOUT_URL Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:LOGOUT_URL</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('LOGOUT_URL')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;LOGOUT_URL.</td>
</tr>
<tr>
<td>Template substitution</td>
<td>#LOGOUT_URL#</td>
</tr>
</tbody>
</table>

**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 describes the supported syntax for referencing PRINTER_FRIENDLY.

Table 2–31  PRINTER_FRIENDLY Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_PRINTER_FRIENDLY (VARCHAR2 DATATYPE)</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('PRINTER_FRIENDLY')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;PRINTER_FRIENDLY.</td>
</tr>
</tbody>
</table>

**PROXY_SERVER**

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

APEX_APPLICATION.G_PROXY_SERVER

**PUBLIC_URL_PREFIX**

PUBLIC_URL_PREFIX is an application-level attribute that identifies a URL to toggle out of a logged in mode to a public view. Table 2–32 describes the supported syntax for referencing PUBLIC_URL_PREFIX.

Table 2–32  PUBLIC_URL_PREFIX Syntax

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:PUBLIC_URL_PREFIX</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('PUBLIC_URL_PREFIX')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;PUBLIC_URL_PREFIX.</td>
</tr>
<tr>
<td>Template substitution</td>
<td>#PUBLIC_URL_PREFIX#</td>
</tr>
</tbody>
</table>

**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 `?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 describes the supported syntax for referencing `REQUEST`.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind variable</td>
<td>:REQUEST</td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_REQUEST</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('REQUEST')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;REQUEST</td>
</tr>
</tbody>
</table>

**Table 2–33 REQUEST Syntax**

**See Also:** "Using REQUEST" on page 2-39

**Using REQUEST**

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

- **About Scope and Value of REQUEST for Posted Pages**
- **About the When Button Pressed Attribute**
- **About Referencing REQUEST Using Declarative Conditions**
- **About Using REQUEST for Show Processing**
- **About Using BRANCH_TO_PAGE_ACCEPT**

**See Also:** "REQUEST" on page 2-38

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

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

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

When you use a button to submit a page, the `REQUEST` value is passed to the page. The Accept processing logic evaluates each validation, process, and branch that uses a When Button Pressed attribute to determine whether the component should run (or fire). When one of these components runs, do not assume that a user actually clicked the associated button and caused the page to be submitted. Keep in mind, that another button using the same request value may have submitted the page. Similarly, JavaScript on the page can also submit the page and pass in a request value.
About Referencing REQUEST Using Declarative Conditions  Many developers reference
REQUEST using conditions. For example, you may want to reset pagination when a user
clicks Go on a report page. You can reset pagination by creating an on-submit page
process. The page process can be made conditional using the condition Request =
Expression 1.

To conditionize an on-submit page process:
1. Under Condition, select the condition type Request = Expression 1.
2. In Expression 1, enter GO.

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

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

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

The following is a similar example using PL/SQL:

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

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

See Also:  Oracle Database Development Guide for information about
developing web applications with PL/SQL and Oracle Database
PL/SQL Packages and Types Reference

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

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

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

SCHEMA OWNER

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

    APEX_APPLICATION.G_FLOW_SCHEMA_OWNER

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

SQLERRM

SQLERRM is a template substitution only available in the Applications Region Error
Message. The following describes the correct syntax for a region template substitution
reference:

    #SQLERRM#
**SYSDATE_YYYYMMDD**

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

- **Bind variable**
  
  :SYSDATE_YYYYMMDD

- **PL/SQL**
  
  `V('SYSDATE_YYYYMMDD')`

- **Direct PL/SQL**
  
  `APEX_APPLICATION.G_SYSDATE (DATE DATATYPE)`

<table>
<thead>
<tr>
<th>Table 2–34</th>
<th>SYSDATE_YYYYMMDD Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Type</td>
<td>Syntax</td>
</tr>
<tr>
<td>Bind variable</td>
<td>:SYSDATE_YYYYMMDD</td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>APEX_APPLICATION.G_SYSDATE (DATE DATATYPE)</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('SYSDATE_YYYYMMDD')</td>
</tr>
</tbody>
</table>

**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 describes the supported syntax for referencing WORKSPACE_IMAGES.

<table>
<thead>
<tr>
<th>Table 2–35</th>
<th>WORKSPACE_IMAGES Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Type</td>
<td>Syntax</td>
</tr>
<tr>
<td>Bind variable</td>
<td>:WORKSPACE_IMAGES</td>
</tr>
<tr>
<td>Direct PL/SQL</td>
<td>Not available</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('WORKSPACE_IMAGES')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;WORKSPACE_IMAGES.</td>
</tr>
<tr>
<td>Template substitution</td>
<td>#WORKSPACE_IMAGES#</td>
</tr>
</tbody>
</table>

**See Also:** "APP_IMAGES" on page 2-30 and "IMAGE_PREFIX" on page 2-37

**WORKSPACE_ID**

Use this substitution string to reference the workspace ID. Table 2–36 describes the supported syntax for referencing WORKSPACE_ID.

<table>
<thead>
<tr>
<th>Table 2–36</th>
<th>WORKSPACE_ID Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Type</td>
<td>Syntax</td>
</tr>
<tr>
<td>Bind variable</td>
<td>:WORKSPACE_ID</td>
</tr>
<tr>
<td>PL/SQL</td>
<td>V('WORKSPACE_ID')</td>
</tr>
<tr>
<td>Substitution string</td>
<td>&amp;WORKSPACE_ID.</td>
</tr>
</tbody>
</table>
Understanding Substitution Strings

Consider the following examples:

- From within an HTML region:
  
  Hello your workspace id is &WORKSPACE_ID.

- Using PL/SQL:
  
  htp.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.
This section describes how to install and run the packaged applications available with Oracle Application Express.

- **About Packaged Applications**
- **Managing Packaged Applications in a Full Development Environment**
- **Understanding Sample Database Application**
- **Understanding AnyCo IT Department Websheet**
- **Using a Packaged Apps Only Workspace**

**See Also:** "Disabling the Creation of Demonstration Objects" and "Disabling the Creation of Sample Websheet Objects" in *Oracle Application Express Administration Guide*

### About Packaged Applications

The Packaged Application gallery includes three types of packaged applications:

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

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

- **Custom applications** - Custom packaged applications are applications you create and that are managed by your instance administrator. Like Oracle-supplied packaged applications, custom packaged application 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 uploads it to make it available. Custom applications are available for installation in a workspace, but are not editable by default.

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

This section describes how to view, install, run, remove, unlock, export, change authentication, and update packaged applications in a full development environment.

- Viewing the Packaged App Gallery
- Viewing the Packaged App Dashboard
- Installing a Packaged Application
- Running an Installed Packaged Application
- Removing an Installed Packaged Application
- Changing Authentication for an Installed Packaged Application
- Unlocking an Installed Productivity Application
- Updating a Packaged Application

See Also: "Managing Custom Applications" in Oracle Application Express Administration Guide

Viewing the Packaged App Gallery

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

To access the Packaged App Gallery:

1. On the Workspace home page, click the Packaged Apps icon.
   
   The Packaged Apps page appears.

2. Click the Packaged App Gallery icon.
   
   The Packaged Apps Gallery appears.
3. To search for an application:
   a. Enter search terms in Search field at the top of the page.
      Search Results appear.
   b. Select an application to view.
      The Packaged Application Details page appears, displaying a summary information about the application. The actual options that display depend upon whether the application has been installed.

4. If the application is not installed, click Install Application to install it.

5. If the application is currently installed, click Manage to change the application authentication or remove the application.

See Also: "Installing a Packaged Application" on page 3-4, "Removing an Installed Packaged Application" on page 3-5, "Changing Authentication for an Installed Packaged Application" on page 3-5, "Unlocking an Installed Productivity Application" on page 3-5, and "Updating a Packaged Application" on page 3-6

Viewing the Packaged App Dashboard

To access the Packaged App Dashboard:

1. On the Workspace home page, click the Packaged Apps icon.
   The Packaged Apps page appears.

2. Click the Dashboard icon.
   The Packaged App Dashboard page appears and displays metrics about packaged applications in the current workspace including the total number of packaged applications, packaged applications not installed, and applications eligible for upgrade.
Installing a Packaged Application

To install a packaged application:

2. Locate the application to be installed.

   **Tip:** Use the Search field at the top of the page to narrow the display.
3. Click an application to install. The Packaged Application Details page appears.
4. Click **Install Application**.
5. Select an Authentication scheme and click **Next**.
6. Click **Install Application** 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 the **Run** icon. See "Running an Installed Packaged Application" on page 3-4.
8. Click **Manage**.

   The Manage dialog displays three buttons:
   - **Change Authentication** - "Changing Authentication for an Installed Packaged Application" on page 3-5.
   - **Unlock** - See "Unlocking an Installed Productivity Application" on page 3-5.

   **Note:** Unlock option is NOT available in Packaged Apps Only Workspace.
   - **Remove** - "Removing an Installed Packaged Application" on page 3-5.

   **Tip:** Once you install a packaged application, it displays under Installed Packaged Apps on the Packaged Apps page.

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. Under Installed Packaged Apps, locate the application to be installed and click the **Run** icon.

   **Tip:** You can also run an installed packaged application from the Packaged App Gallery or from Application Builder.
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" on page 2-10

### 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 **Packaged Apps** icon.
3. Locate the Installed Packaged Apps region.
4. Locate the application and click the **Manage** icon.
5. Click **Remove**.
6. When prompted, click **Remove Application**.

**Tip:** You can also perform this task on the Application Details page. See "Viewing the Packaged App Gallery" on page 3-2.

### 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. Locate the Installed Packaged Apps region.
3. Locate the application and click the **Manage** icon.
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 the Packaged App Gallery" on page 3-2.

**See Also:** "Establishing User Identity Through Authentication" on page 20-37

### Unlocking an Installed Productivity Application

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

**Note:** Unlocking a packaged application makes it ineligible for future upgrades or support by Oracle Support.

To unlock an installed packaged application:

1. On the Workspace home page, click the **Packaged Apps** icon. The Packaged Apps page appears.
Understanding Sample Database Application

2. Locate the Installed Packaged Apps region.
3. Locate the application and click the Manage icon.
4. Click Unlock.
   The Unlock Packaged Application page appears.
5. Click Unlock Application.

See Also: "Installing a Packaged Application" on page 3-4

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:

   The Packaged Apps page appears.
2. Locate and click the application to be updated.
   The Packaged Application Details page appears.
3. Click Update Application.
   The Update Packaged Application page appears.

Note: Because sample applications are unlocked by default, they cannot be updated. If you wish to access the Universal theme look and feel available in Oracle Application Express Release 5.0, reinstall the application.

See Also: "Running an Installed Packaged Application" on page 3-4

Understanding Sample Database Application

This section describes the sample packaged application, 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
- Understanding Sample Database Application
- Modifying Sample Database Application

Installing and Running Sample Database Application

Tip: The Sample Database Application is automatically installed in new full workspace.

To install Sample Database Application:

2. Locate the Sample Database Application.

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

3. Click the Sample Database Application application image.

   The Packaged Application Details page appears.

4. Click Install Application.

5. Select an Authentication scheme and click Next.

6. Click Install Application again.

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

   **Tip:** Once you install a packaged application, it displays under Installed Packaged Apps on the Packaged Apps page.

7. Click Run.

8. Enter the appropriate login credentials:
   a. Username - Enter your workspace username.
   b. Password - Enter your workspace password.
   c. Click Sign In.

**Understanding Sample Database Application**

*Sample Database Application* demonstrates the following functionality:

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

This section describes the functionality built-in to *Sample Database Application*.
Accessing the Mobile Version

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.

See Also: "What Is an Application Page?" on page 2-4

Home Page

The home page contains the following main regions:

- **Dashboard** - Demonstrates the use of a Badge List plug-in. This regions 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 Badge List plug-in. This regions 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** - Demonstrates the Slider Tooltip plug-in. 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 additional details for a given row by hovering over it to display a detailed tooltip, or by clicking the customer name or by clicking the View Customer 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.
Customers Page
The Customers page enables users to view and edit customer information.

Customers is an interactive report for 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: "Customizing Interactive Reports as an End User" on page 10-57 and "About the Actions Menu" on page 10-58

Using the Customers Page
This section describes functionality built into the Customers page.

- Editing a Customer
- Creating a Customer
- Uploading Data

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.

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.

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 at the bottom of the window.
      – Copy and Paste. If selected, a Copy and Paste field displays at the bottom
        on the window.

   Tip: The option you select determines what additional fields display.
   To view sample data, expand the Sample Data region.

   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. To view sample data,
      expand the Sample Data region.
   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.

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.

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 as an End User" on page 10-57 and "About the Actions Menu" on page 10-58

### Using the Products Page
This section describes functionality built into the Customers page.

- **Editing a Product**
- **Creating a Product**

**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**.

**Creating a Product** To add a new product:
1. Click the **Create Product** button.
2. On the Product Details page, edit the fields and click **Add Product**.

### Orders Page
The Orders page enables users to view and edit customer orders.

**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:** "Customizing Interactive Reports as an End User" on page 10-57 and "About the Actions Menu" on page 10-58

### Using the Orders Page
This section describes functionality built into the Orders page.

- **Editing an Order**
Understanding Sample Database Application

Creating an Order

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.

Creating an Order  To create an order:
1. Click the Enter New Order button.
   The Enter New Order wizard appears.
2. Follow the on-screen instructions to enter order information.
3. Click Place Order.

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" on page 12-22, "Creating Maps" on page 12-12, and "Creating Trees" on page 15-49

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. To learn more, see "Managing Feedback" on page 5-16.
- **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. See "Using Theme Styles" on page 13-3.

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.

**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*. See "Running an Installed Packaged Application" on page 3-4.
2. Click **Mobile** in the upper right corner.
   
   The Mobile version of *Sample Database Application* appears.
Understanding Sample Database Application

3. To view a page, select it.

4. To edit a specific customer or product, select it.
   A form appears.

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 Application Builder.

Modifying Sample Database Application

This section describes how to modify a sample database application.

- Why Modify a Sample Database Application?
- About Using the Runtime Developer Toolbar
- Editing Sample Database Application

**See Also:** "Managing Packaged Applications in a Full Development Environment" on page 3-2

**Why Modify a Sample Database Application?**

Once you understand the type of functionality available in a packaged application, such as *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.

About Using the Runtime Developer Toolbar

When you run a desktop application from Application Builder, 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. The Runtime Developer toolbar only displays if the Availability, Status attribute on the Application Definition page is set to **Available with Developer Toolbar**.

The Runtime Developer toolbar consists of the following controls:

- **Home** links to the Workspace home page. See "Understanding the Workspace Home Page" on page 1-10.
- **Application** links to the Application home page. See "Using the Application Builder Home Page" on page 4-2.
- **Edit Page** displays the currently running page ID. The View that appears depends upon your current preference. See "Understanding the Difference Between Page Designer and Component View" on page 2-4.
- **Session** links you to session state information for the current page. See "Viewing Session State" on page 2-15.
- **View Debug** displays the Debug reports. See "Debugging an Application" on page 22-1.
- **Debug** toggles the page between **Debug** and **No Debug** mode. See "Utilizing Debug Mode" on page 22-3.
- **Show Grid** toggles between **Show Grid** and **Hide Grid**. This option only applies if a grid layout is used and that layout supports showing a grid and the page has more than one column.
- **Quick Edit** enables developers to enter Quick Edit mode and then select the desired component to instantly access that component in Page Designer. Press ESCAPE or click outside a component to exit quick edit mode.
- **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" on page 13-3.
- **Developer Toolbar Options** displays on the far right and resembles a mechanical gear. You can customize how the Developer Toolbar displays by using the following controls:
  - **Auto Hide**
  - **Show Icons Only**

Note: The Runtime Developer toolbar does not display on locked packaged applications pages or on jQuery Mobile Smartphone pages.
Display Position (Top, Left, Bottom, Right)

Note: The Developer toolbar does not display on locked packaged applications pages or on jQuery Mobile Smartphone pages.

See Also: "Runtime Developer Toolbar" on page 8-24

Editing Sample Database Application

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.

Click the following icons to run the current application, edit supporting objects, create shared components, or export and import an application:

- **Run Application** submits the pages in the current application to the Application Express engine to render viewable HTML. See "How the Application Express Engine Renders and Processes Pages" on page 2-12.

- **Supporting Objects** links to the Supporting Objects page. See "How to Create a Custom Packaged Application" on page 24-6.

- **Shared Components** links to a list of shared components and user interface controls that can display or be applied on every page within an application. See "Managing Shared Components" on page 18-1.

- **Utilities** links to the Utilities page. Use this page to monitor developer activity, view dashboards, run Advisor, and view numerous other reports. See "Using Application Builder Utilities" on page 7-61.

- **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 "Exporting an Application and Application Components" on page 24-12.

The pages that comprise the application appear on 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 Page field and click Go.

See Also: "Using Application Home Page" and "Using Component View" on page 9-5
Understanding AnyCo IT Department Websheet

This section describes the packaged application, *Sample Websheet - AnyCo IT Department*. *Sample Websheet - AnyCo IT Department* highlights the features and capabilities of Websheets.

- Installing and Running AnyCo IT Department
- About Websheet Navigation
- Understanding AnyCo IT Department
- Modifying Websheet Content

**See Also:** "Creating Websheet Applications" on page 6-1 and "Using Websheets" in Oracle Application Express End User Guide

Installing and Running AnyCo IT Department

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

2. Locate the *Sample Websheet - AnyCo IT Department*. A summary page appears.
3. Click **Install Application**.
4. Select an Authentication scheme and click **Next**.
5. Click **Install Application**. A success message appears.
6. To run the application, click the **Run** icon.
7. Enter the appropriate login credentials:
   a. Username - Enter your workspace username.
   b. Password - Enter your workspace password.
   c. Click **Sign In**.

The home page displays in a new window.

**See Also:** "How Your Browser Impacts the Way Applications Run" on page 2-10

About 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 hide and unhide a page section by clicking the arrow to the left of the Section title. In the following example, Systems Overview is hidden.
Using Breadcrumbs to Navigate Between Websheet Pages

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

Understanding AnyCo IT Department

Sample Websheet - AnyCo IT Department demonstrates the following functionality:

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

The following sections describe specific functionality available on each page.

- Understanding Navigation
- Home Page
- Project Review Page
- Systems Page
- Planned Purchase Review
- Systems Budget Review

**Understanding Navigation**

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 link to another page, select the page name.
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.
Projects Page

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

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

Systems Page

The Systems page contains the following sections:

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

System is a navigation section that includes links to the sections, Systems Overview, Systems Maintenance, and Purchasing Projections.

System Overview is a text section that includes an embedded image and links to other pages.

System 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.

Planned Purchase Review

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.
Systems Budget Review

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.

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?
- Updating a Websheet Section

**Why Modify Sample Websheet Application?**

Analyzing the deconstruction of an application is an efficient way to speed up the learning process. If you break something, you can quickly delete the demonstration application and install it again. See "Managing Packaged Applications in a Full Development Environment" on 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.

The section that follows offers a high-level discussion of editing a Websheet.

See Also: "Creating Websheet Applications" on page 6-1

Updating a Websheet Section

To update a Websheet section:

1. Run the Websheet as described in "About Packaged Applications" on page 3-1.
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**.

**Using a Packaged Apps Only Workspace**

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

This section describes how to request and use a Packaged Apps Only workspace.
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 Application Builder or SQL Workshop. A Packaged Apps Only workspace includes the Packaged App Gallery, Packaged App Dashboard, 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.

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. Follow the on-screen instructions.

Tip: You can manually control whether a workspace includes development tools, such as Application Builder, by disabling access to development components. See "Configuring Workspace Preferences" and "Managing Component Availability for an Instance" in Oracle Application Express Administration Guide.

Signing In To a Packaged Apps Only Workspace

To sign in to a Packaged Apps Only workspace:

1. Sign in to the Packaged Apps Only workspace. The home page appears.
2. Click an icon:
   - Dashboard - The Packaged App Dashboard in a Packaged Apps Only workspace is the same as what displays in a full development environment. See "Viewing the Packaged App Dashboard" on page 3-3.
Using the Packaged App Gallery

To access the Packaged App Gallery:

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

   **Tip:** Sample packaged applications are not available in a Packaged App Only workspace. Otherwise, the Packaged App Gallery in a Packaged Apps Only workspace works the same as in a full development environment. See "Managing Packaged Applications in a Full Development Environment" on page 3-2.

Utilizing Packaged App Administration

This section describes functionality available on the Packaged App Administration page.

- **Making a Service Request**
- **Setting Workspace Preferences**
- **Managing Users**
- **Managing Saved Interactive Reports**
- **Managing Interactive Report Subscriptions**
- **Viewing Activity Reports**
- **Viewing Page Views**
- **Viewing Top Users**

**See Also:** "Workspace and Application Administration" in *Oracle Application Express Administration Guide*

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 the Packaged Apps Only workspace.
2. Click **Packaged App Administration**.
3. Click Make a Service Request.

4. For 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).

5. Follow the on-screen instructions.

**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 the Packaged Apps Only workspace.
2. Click Packaged App Administration.
3. Click Set Workspace Preferences.
4. Under Account Login Control:
   - 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.
   - 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.
   - 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.
5. Click Apply Changes.

**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*

---

**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 the Packaged Apps Only workspace.
2. Click **Packaged App Administration**.
3. 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*

---

**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 the Packaged Apps Only workspace.
2. Click **Packaged App Administration**.
3. 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*

---

**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 App Administration**.
3. Click **Application Activity**.
**Viewing Page Views**
Click **Page Views** to view an interactive report of each view and detailing the user, elapses time, and page.

To view the Page View report:
1. Sign in to Oracle Application Express.
2. Click **Packaged App Administration**.
3. Click **Page Views**.

**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 App Administration**.
3. Click **Top Users**.
Developers use Application Builder to build the pages that make up an Oracle Application Express application. This section provides important background information about the Application Builder user interface.

- Accessing Application Builder
- Using the Application Builder Home Page
- Accessing Application Home Page
- Using Application Home Page

See Also: "Quick Start" on page 1-1 and "Application Builder Concepts" on page 2-1

Accessing Application Builder

To access Application Builder:

2. Click the Application Builder icon. The Application Builder home page appears.
Using the Application Builder Home Page

This section describes the Application Builder home page. The Application Builder home page displays all installed applications in the current Oracle Application Express instance.

- Application Builder Home Page Navigation Bar
- Reset and Create Buttons
- Application Builder Home Recent List
- Migrations Link

See Also: "Accessing Application Builder" on page 4-1 and "Developer Navigation Tools" on page 1-13

Application Builder Home Page Navigation Bar

You can customize the appearance of the Application Builder home page using the navigation bar at the top of the 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.
Accessing Application Home Page

- **Actions menu** - Displays the Actions menu. Use this menu to customize the report view. See “About the Actions Menu” on page 10-58.

**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. See “About Running the Create Application Wizard” on page 2-2.

**Application Builder Home Recent List**

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

**Migrations Link**

The Migrations Link 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

**Accessing Application Home Page**

To access the Application home page:

1. Sign in to Oracle Application Express.
   The Workspace home page appears.

2. Click the Application Builder icon.
   The Application Builder home page appears.

3. Select an application on the Application home page.
   The Application home page appears.
Using Application Home Page

When you select an application, the Application home page appears. Use the Application home page to run, edit, import, export, copy, or delete applications.

- Edit Application Properties Button
- Application Home Page Icons
- Create Page Button
- Application Home Page Navigation Bar
- About Customizing the Application Home Page
- Application Home Page Tasks List
- Recently Edited Pages

See Also: "Using Application Home Page" on page 4-4

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 the Application Attributes" on page 7-14
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. See "How the Application Express Engine Renders and Processes Pages" on page 2-12.

- **Supporting Objects** links to the Supporting Objects page. See "How to Create a Custom Packaged Application" on page 24-6.

- **Shared Components** links to a list of shared components and user interface controls that can display or be applied on every page within an application. See "Managing Shared Components" on page 18-1.

- **Utilities** links to the Utilities page. Use this page to monitor developer activity, view dashboards, run Advisor, and view numerous other reports. See "Using Application Builder Utilities" on page 7-61.

- **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 "Exporting an Application and Application Components" on page 24-12.
Create Page Button

Clicking the Create Page button launches a wizard that walks you through creating a page in an application. See "Creating a Page from the Application Home Page" on page 7-40.

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 “About the Actions Menu” on page 10-58.

  **See Also:** "Application Builder Home Page Navigation Bar" on page 4-2

### About Customizing the Application Home Page

You can control how the Application home page appears by selecting the following icons: View Icons, View Report, or View Detail.

**View Icons** (the default) display each page as an icon.

<table>
<thead>
<tr>
<th>Page</th>
<th>Name</th>
<th>Updated</th>
<th>By</th>
<th>Page Type</th>
<th>Group</th>
<th>User Interface</th>
<th>Lock</th>
<th>Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Page Zero</td>
<td>7 weeks ago</td>
<td>allan</td>
<td>Global Page</td>
<td>Desktop</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Sample Database Application</td>
<td>4 days ago</td>
<td>-</td>
<td>Home</td>
<td>Desktop</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Customers</td>
<td>4 days ago</td>
<td>-</td>
<td>Interactive Report</td>
<td>Desktop</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Products</td>
<td>4 days ago</td>
<td>-</td>
<td>Interactive Report</td>
<td>Desktop</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Report** mode 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, and lock status. To view a page, click the page ID. 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 pages cannot be run directly.
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. See "Deleting an Application" on page 7-13.

- **Copy this Application** creates a copy of the current application. See "Copying a Database Application" on page 7-8.

**See Also:**  "Using Application Home Page" on page 4-4

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.
Managing the Application Life Cycle with Team Development

Team Development is a built-in development management tool that 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.

- Team Development Home Page
- About Utilizing the Tags Summary
- Tracking Features
- Tracking Milestones
- Tracking To Dos
- Managing Bugs
- Managing Feedback
- Managing Team Development Utilities

See Also: "Managing Team Development” in Oracle Application Express Administration Guide

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” on page 5-1

Team Development Home Page

The Team Development home page functions as a starting point for managing features, milestones, To Dos, bugs, and feedback.
The following large icons display on the Team Development home page:

- **Milestones.** Track events associated with the development process. You can associate milestones with features, bugs, and to dos. See "Tracking Milestones" on page 5-8.

- **Features.** Track features from initial concept through implementation. You can organize features by release, assignee, tags, or associated milestones. See "Tracking Features" on page 5-3.

- **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. See "Tracking To Dos" on page 5-10.

- **Bugs.** Track software defects or bugs. Bugs can be assigned, associated with milestones, and tracked by due date, status, and other attributes. See "Managing Bugs" on page 5-13.

- **Feedback.** Gathering real-time comments, enhancement requests, and bugs from your application users. See "Managing Feedback" on page 5-16.

**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. See "Managing News Entries" on page 5-22.

**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. See "Accessing Team Development Utilities" on page 5-19.

**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. See "Tracking Milestones" on page 5-8.
Tracking Features

- **Bugs.** Provides a summary of closed bugs. See "Managing Feedback" on page 5-16.
- **Features.** Lists a percentage of features that are functionally complete. See "Tracking Features" on page 5-3.
- **To Dos.** Lists a percentage of completed To Dos. See "Tracking To Dos" on page 5-10.
- **Feedback.** Displays recent feedback entries. See "Managing Feedback" on page 5-16.
- **Team Development Summary.** Lists the total number of features, to dos, milestones, bugs, and feedback entries.
- **Tags.** See "About Utilizing the Tags Summary" on page 5-3.

**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.

**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).

**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**
- **Creating a Feature**
- **Updating a Feature**
- **How to Attach Files to a Feature**
- **Viewing Other Feature Reports**

**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 Assignee, Release, 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 as an End User" on page 10-57 and "Updating a Feature" on page 5-4

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. See "Viewing Field-Level Help" on page 1-21.
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" on page 5-21

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" on page 5-6

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" on page 5-21

How to Attach Files to a Feature

If enabled at the workspace-level, you can attach files to features.
Tracking Features

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. See "Updating a Feature" on page 5-4.
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.

Viewing, Editing, and Removing Attached Files

To view, edit, or remove attached files:
1. Navigate to the Feature page. See "Updating a Feature" on page 5-4.
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.

Viewing Other Feature Reports

This section describes feature reports available in Team Development.

- Viewing the Features Tree
- Viewing the Features Calendar
- Viewing The Feature History Report
- Viewing the Feature Progress Log
- Viewing the Feature Focus Areas Chart
- Viewing the Feature Owners Chart
- Feature Utilities

See Also: "Viewing File Attachments" on page 5-22 and "Enabling Support for File Upload in Team Development" in Oracle Application Express Administration Guide.
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.

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.

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 as an End User" on page 10-57 and "About the Actions Menu" on page 10-58

**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 as an End User" on page 10-57 and "About the Actions Menu" on page 10-58

**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" on page 5-21

**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**.
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” on page 5-19

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
- Creating a Milestone
- Updating a Milestone
- Viewing Other Milestone Reports

Viewing the Milestones Report
To view milestones:

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

   The Milestones Dashboard appears.

   The Milestones Dashboard displays an overview of milestones within the workspace. You can filter by release or link to a specific milestone or milestones by owner.

3. Click the Report tab.

   The Milestones Report page displays as an interactive report. You can customize the appearance of page, using the Search bar at the top of the page.

   See Also: "Customizing Interactive Reports as an End User" on page 10-57 and "Updating a Milestone" on page 5-9

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 and displays the following regions:

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. See “Viewing Field-Level Help” on page 1-21.
5. To add new values to editable select lists, enter a value in the field with the title New. For example, to add name to the Owner list, enter a value in New Owner. The value you enter is added.

6. Click Create Milestone.

Updating 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. See "Viewing Field-Level Help" on page 1-21.
6. Click Apply Changes.

Viewing Other Milestone Reports

This section describes milestone reports available in Team Development.

- Viewing the Milestone Calendar
- Viewing the Milestones By Owner Report
- Viewing the Features by Milestone Report

See Also: "Accessing Team Development Utilities" on page 5-19

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.

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**.

**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 as an End User" on page 10-57 and "About the Actions Menu" on page 10-58

**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
- Creating a To Do
- Updating a To Do
- How to Attach Files to a To Do
- Viewing Other To Do Reports

**Viewing To Dos**
To view To Dos:
1. On the Workspace home page, click the **Team Development** icon.
2. Click the **To Dos** icon.
   The To Dos Dashboard appears. The To Dos Dashboard displays an overview of To Dos within the workspace. You can filter by release, assignee, or application.
3. Click the **Report** tab.
   The To Dos Report page appears.
   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 as an End User" on page 10-57 and "Updating a To Do" on page 5-11
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. See "Viewing Field-Level Help" on page 1-21.
5. To add new values to editable select lists, enter a value in the field with the title New. For example, to add name to the Assigned To list, enter a value in New Assignee.
   The value you enter is added.
6. Click Create To Do.

Updating a To Do

To update a To Do:

1. On the Workspace home page, click the Team Development icon.
2. Click the To Dos icon.
   The To Dos Dashboard appears.
3. Click the Report tab.
4. Select a milestone.
5. Edit the appropriate fields.
6. Click Apply Changes.

How to Attach Files to a To Do

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

- Attaching a File to a To Do
- Viewing, Editing, and Removing Attached Files

See Also: "Viewing File Attachments" on page 5-22 and "Enabling Support for File Upload in Team Development" in Oracle Application Express Administration Guide.

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. See "Updating a To Do" on page 5-11.
2. From Actions, click Add File.

3. On Add File:
   a. File - Locate the file to be uploaded.
   b. Description - Enter a description of the file.
   c. Click Add File.

   A new region, called Files, displays at the bottom of the page.

**Viewing, Editing, and Removing Attached Files**

To manage attached files:

1. Navigate to the To Do page. See "Updating a To Do" on page 5-11.
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.

**Viewing Other To Do Reports**

This section describes To Do reports available in Team Development.

- Viewing the To Do Calendar
- Viewing the To Do Progress Log

**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 Do 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.

**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 as an End User" on page 10-57 and "About the Actions Menu" on page 10-58

### Managing Bugs

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

- **Viewing Bugs**
- **Creating a Bug**
- **Editing a Bug**
- **How to Attach Files to a Bug**
- **Viewing Other Bug Reports**

### Viewing Bugs

To view bugs:

1. On the Workspace home page, click the **Team Development** icon.
2. Click the **Bugs** icon.
   
   The Bugs Dashboard appears. The Bugs Dashboard displays an overview of known software defects.
3. Click the **Report** tab.
   
   The Bugs Report page displays an interactive report of all bugs. You can customize the appearance of the page using the Search bar at the top of the page. To edit a bug, click the **Edit** icon.

**See Also:** "Customizing Interactive Reports as an End User" on page 10-57 and "Editing a Bug" on page 5-14

### 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. See "Viewing Field-Level Help" on page 1-21.
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.

**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.
3. Click the Report tab.
   The Bugs Report page appears.
4. Select a bug.
5. Edit the appropriate fields.
   To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
6. Click Apply Changes.

**How to Attach Files to a Bug**

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

- Attaching a File to a Bug
- Viewing, Editing, and Removing Attached Files

**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. Navigate to the bug. See "Editing a Bug" on page 5-14.
2. From Actions, click Add File.
3. On Add File:
   a. File - Locate the file to be uploaded.
   b. Description - Enter a description of the file.
   c. Click Add File.
      A new region, called Files, displays at the bottom of the page.

**Viewing, Editing, and Removing Attached Files**

To manage attached files:
1. Navigate to the Bug page. See "Editing a Bug" on page 5-14.
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.

**Viewing Other Bug Reports**

This section describes bug reports available in Team Development.

- Bugs Dashboard
- Viewing the Bug Calendar
- Viewing the Bugs By Developer
- Viewing the Bugs By Day

**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.

**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.

**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**.

**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**.

**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**
- **Adding a Feedback Page to an Application**
- **Reviewing Feedback**
- **Viewing Other Feedback Reports**

**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 as an End User" on page 10-57

**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 **Application 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. See "Viewing Field-Level Help” on page 1-21.

   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 the Application Attributes” on page 7-14

6. Click Create.

**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**.

**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. See “Viewing Field-Level Help” on page 1-21.
7. To delete a feedback entry, click **Delete**.
8. To submit your changes, click **Apply Changes**.

**Viewing Other Feedback Reports**

This section describes feedback reports available in Team Development.

- **Viewing the Feedback Calendar**
- **Viewing Feedback By Filing User**

**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.
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.

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
- Configuring Team Development Settings
- Viewing a Release Summary
- Managing Feature Utilities
- Editing Focus Areas
- Viewing File Attachments
- Purging Data
- Managing News Entries
- Managing Links

See Also: "Managing Team Development" in Oracle Application Express Administration Guide

Accessing Team Development Utilities
Use Team Development Utilities to manage team development data. You can access Team Development Utilities from either the Utilities region on the Team Development home page or from Utilities page.

To access Team Development Utilities:
1. On the Workspace home page, click the Team Development icon.
2. Note the Utilities region on the right side of the page.
   Tip: You can access most utilities by either clicking links in the Utilities region or by accessing the Utilities page.
3. From the Utilities region on the right side of the page, click All Utilities.
4. The Utilities page features the following links:
   - Team Development Settings - See "Configuring Team Development Settings" on page 5-20.

■ **Enable Files** - Links to the administration page that enables you to add support for file upload. See "Enabling Support for File Upload in Team Development" in *Oracle Application Express Administration Guide*.

■ **Feature Utilities** - Use the Feature Utilities page to 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. See “Editing Focus Areas” on page 5-21.

■ **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. See "Viewing File Attachments" on page 5-22.

**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** - Deletes all entries for the selected components. This is useful when beginning a new development cycle or after using a workspace for testing.


■ **Manage Links** - Manage links to share with other workspace users. See "Managing Links" on page 5-23.

■ **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.

### Configuring Team Development Settings

Team Development Settings enable you to 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. See "Viewing Field-Level Help" on page 1-21.
4. Click **Apply Changes**.
   
   **Tip:** Click **Reset to Default** to return the settings to the default values.

**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 (*).
   c. To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
5. Click **Email**.
   
   **Tip:** Click **Reset** to return the report to the default view.

**Managing Feature Utilities**

Use the Feature Utilities page to assign milestones to features, set feature due dates to milestone dates, 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.

**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**.

**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*.

**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**.

**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.

Managing Links

Use the Links page to share links within Team Development.

- Adding Links to the Links Page
- Editing or Deleting a Link

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 (*)
   c. To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
   d. Click Create Link.

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**.
Creating Websheet Applications

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

- Creating a Websheet Application
- Copying an Existing Websheet Page
- Running a Websheet
- Editing Page Details
- Deleting a Websheet Application
- Viewing the Websheet Dashboard
- Monitoring Websheet Activity
- Understanding Websheet Properties
- Controlling Websheet Access

See Also: "Using Websheets" in Oracle Application Express End User Guide and "Removing and Validating Websheet Database Objects" and "Disabling the Creation of Sample Websheet Objects" in Oracle Application Express Administration Guide

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?" on page 2-2, "Understanding AnyCo IT Department Websheet" on page 3-17, and "Using Websheets" in Oracle Application Express End User Guide

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 Application Builder
Creating a Websheet Application from a Running Websheet

See Also: "Using Websheets" in Oracle Application Express End User Guide and "Removing and Validating Websheet Database Objects" and "Disabling the Creation of Sample Websheet Objects" in Oracle Application Express Administration Guide

Creating a Websheet from Application Builder

To create a Websheet application:

1. On the Workspace home page, click the Application 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:
      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.

   Your Websheet application displays in a new window.
7. Add Websheet content.

See Also: "Using Websheets" in Oracle Application Express End User Guide and "About Publishing the Websheet Application URL" on page 24-30
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" on page 6-3.
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: "Using Websheets” in Oracle Application Express End User Guide and "About Publishing the Websheet Application URL” on page 24-30

Copying an Existing Websheet Page

To copy an existing Websheet page:

1. Run the Websheet application. See "Running a Websheet” on page 6-3.
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” on page 6-5.

Running a Websheet

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

■ Running a Websheet from the Application Builder Home Page
■ Running a Websheet from the Application Home Page
■ Determining a Websheet URL
Running a Websheet from the Application Builder Home Page

To run a Websheet from the Application Builder home page:

1. On the Workspace home page, click the Application Builder icon.
2. From the Search bar, click the View Report icon.
3. Locate the Websheet application in the Applications list.
4. Click the Run icon in the far right column.
5. Enter your Websheet credentials:
   a. Username - Enter your username.
   b. Password - Enter your password.
   c. Click Log In or Sign In.

   **Tip:** The language that displays differs depending upon who creates the application. For Oracle-supplied applications, including 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.

Running a Websheet from the Application Home Page

To run an entire application from the Application home page:

1. On the Workspace home page, click the Application Builder icon.
   The Application Builder home page appears.
2. Select the Websheet application.
3. Click the Run button.
4. Enter your Websheet credentials:
   a. Username - Enter your username.
   b. Password - Enter your password.
   c. Click Log In or Sign In.

   **Tip:** The language that displays differs depending upon who creates the application. For Oracle-supplied applications, including 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.
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” on page 24-30.

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. See "Running a Websheet" on page 6-3.
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.

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 Application Builder icon. The Application Builder home page appears.
2. Select the Websheet application. The Websheet Properties page appears.
3. Click Delete.

**See Also:** "Understanding Websheet Properties” on page 6-6

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” on page 6-3.
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.

**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. See "Running a Websheet” on page 6-3.
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 as an End User” on page 10-57.

**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**
- **Websheet Properties**

**See Also:** "Deleting a Websheet Application” on page 6-5
Editing Websheet Properties

To access the Websheet Properties page:

1. Log in to and run the Websheet. See “Running a Websheet” on page 6-3.
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:**  "Websheet Properties" on page 6-7

Websheet Properties

This section describes the attributes on the Websheet Properties page.

- **Websheet**
- **Details**
- **Logo**
- **Style**
- **Authentication**
- **Authorization**
- **SQL**

**Websheet**

Use the Websheet section to control general Websheet application properties. Table 6-1 describes Application attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Identifies the name of the application. This name displays within the breadcrumb when the application is running.</td>
</tr>
<tr>
<td>Status</td>
<td>Identifies the status of the application.</td>
</tr>
<tr>
<td>Home Page</td>
<td>Select the home page for this Websheet. This is the first page a user sees when running the Websheet.</td>
</tr>
<tr>
<td>Application Date Format</td>
<td>Determines the date format to be used in the application.</td>
</tr>
<tr>
<td></td>
<td>This date format is used to alter the <code>NLS_DATE_FORMAT</code> 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.</td>
</tr>
<tr>
<td>Default Application Language</td>
<td>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.</td>
</tr>
<tr>
<td>Default Application Territory</td>
<td>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.</td>
</tr>
</tbody>
</table>
Details
Use the Details section to edit the application description and login message. Table 6–2 describes Details attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Identifies the user who owns the application.</td>
</tr>
<tr>
<td>Description</td>
<td>An optional description that describes the application.</td>
</tr>
<tr>
<td>Login Message</td>
<td>Display this message on the login page.</td>
</tr>
</tbody>
</table>

Logo
Use the Logo section to create a logo for this Websheet application. This logo displays at the top of the Websheet. Table 6–3 describes Logo attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logo Type</td>
<td>Select the type of logo being used for the application.</td>
</tr>
<tr>
<td>Logo</td>
<td>Enter a full text string to use as application logo.</td>
</tr>
</tbody>
</table>

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

Authentication
Use the Authentication section to select an authentication scheme for this Websheet. Table 6–4 describes Authentication attributes.
Authorization

Use the Authorization section to control who can log in to a Websheet application. Table 6–5 describes Application attributes.

### Table 6–4 Authentication

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Select the way users of the application are authenticated. To change the authentication scheme, click Edit Authentication. The following list displays:</td>
</tr>
<tr>
<td></td>
<td>- Application Express Account - Authenticate using developer account username and password.</td>
</tr>
<tr>
<td></td>
<td>- Single Sign-On - Authenticate using Oracle Application Server Single Sign-On. This must be configured for the Application Express instance.</td>
</tr>
<tr>
<td></td>
<td>- LDAP - Authenticate using an LDAP directory. Specify additional parameters on this page to configure this.</td>
</tr>
<tr>
<td></td>
<td>- Custom - Provide your own code for authentication and session management tasks. See &quot;Controlling Websheet Access&quot; on page 6-10.</td>
</tr>
<tr>
<td>Logout URL</td>
<td>Specify a URL to become this application's Logout URL attribute. This attribute is referenced in logout links as follows: &amp;LOGOUT_URL.</td>
</tr>
<tr>
<td></td>
<td>If the application uses built-in session verification, you may use &quot;www_flow_custom_auth_std.ws_logout?p_websheet_app_id=&amp;WS_APP_ID.&amp;p_next_url=ws?p=WS_APP_ID.&amp;home&quot; to invoke a built-in logout procedure.</td>
</tr>
<tr>
<td>Cookie Name</td>
<td>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.</td>
</tr>
<tr>
<td>Cookie Path</td>
<td>Path attribute for the session cookie.</td>
</tr>
<tr>
<td>Cookie Domain</td>
<td>Domain attribute for the session cookie.</td>
</tr>
<tr>
<td>Secure</td>
<td>Options include:</td>
</tr>
<tr>
<td></td>
<td>- Yes - Allow the session management cookie to be sent from the browser only when the protocol is HTTPS.</td>
</tr>
<tr>
<td></td>
<td>- No - Allow the session management cookie to be sent from the browser when the protocol is HTTP or HTTPS.</td>
</tr>
</tbody>
</table>

### Table 6–5 Authorization

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Control List Type</td>
<td>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.</td>
</tr>
<tr>
<td>Allow Public Access</td>
<td>Options include:</td>
</tr>
<tr>
<td></td>
<td>- Yes - Enable users to run the application in a read-only mode without requiring users to log in.</td>
</tr>
<tr>
<td></td>
<td>- No - All users must log in to run the application.</td>
</tr>
</tbody>
</table>
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" on page 6-10.

**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*.

### Controlling Websheet Access

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

- **How Websheet Access Works**
- **Specifying the Access Control List Type**
- **Creating Access Control List Entries**

### How Websheet Access Works

An administrator controls who can log in to a Websheet application by specifying one of the following authentication mechanisms:

- Public access (no authentication)
- Oracle Application Express Account
- Single Sign-On Verification
- LDAP Credentials Verification
- Custom

If authentication is defined as **Custom**, an access control list manages the privileges of this application's users.

If authentication is **Oracle Application Express Account**, you must specify whether authorization is determined by the user role defined in the Application Express workspace, or if authorization is determined by the access control list. For all authentication types other than Oracle Application Express Accounts, authorization is always determined by the access control list.

Once a user logs in to an application, a user's assigned role determines access. User assigned roles include:

- **Reader** - May view, but not edit content.
- **Contributor** - May view and edit content.
- **Administrator** - May view and edit content and edit Websheet Properties and manage the Access Control List.
To learn more about user roles and privileges, click Help and then select Access Control.

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

   See Also: "Authentication" on page 6-8

**Creating Access Control List Entries**

To create access control list entries:
1. Log in to and run the Websheet. See "Running a Websheet" on page 6-3.
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.
Creating Database Applications

This section describes how to use Application Builder to create database applications. It explains the advantages of using the Create Application Wizard, how to create database applications, how to edit application attributes, provides an overview of the available views for managing pages, explains how to manage application comments, and provides a summary of application utilities.

- Using the Create Application Wizard
- Creating a Database Application
- Creating Mobile Applications
- Deleting an Application
- Managing Application Attributes
- Managing Pages in a Database Application
- Using the Find Icon
- Creating Page Groups
- Locking and Unlocking a Page
- Adding Developer Comments
- Adding Database Application Comments
- Managing Application Groups
- Accessing Application Utilities

See Also: "Managing User Interface Design" on page 2-8, "Utilizing Packaged Applications" on page 3-1, "What is the Difference Between a Database and Websheet Application?" on page 2-2, "Understanding Application Builder" on page 4-1, "Developing Reports" on page 10-1, "Controlling Page Layout" on page 14-1, and "Adding Navigation" on page 16-1

Using the Create Application Wizard

The Create Application Wizard enables you to build a complete application containing multiple pages. This section explains how the Create Application Wizard works and describes the types of pages you can create when running it.

- About the Create Application Wizard
- Available Page Types When Running the Create Application Wizard
About the Create Application Wizard

The Create Application Wizard enables you 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 reports, interactive reports, forms, tabular forms, and master detail 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 a tabular form. 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.

Available Page Types When Running the Create Application Wizard

The User Interface you select when running the Create Application Wizard determines the available page types. Table 7–1, "Create Application Wizard - Available Page Types" lists the different types of pages you can create when running the Create Application Wizard. Not all page types are supported by all user interfaces.

See Also: "Available Page Types When Running the Create Application Wizard" on page 7-2, "About Adding Pages to a Database Application" on page 7-36, "Building Queries with Query Builder" in Oracle Application Express SQL Workshop Guide and "Leveraging Application Builder Defaults" in Oracle Application Express Administration Guide

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>Creates a page with no built-in functionality.</td>
<td>Available</td>
<td></td>
<td>Available</td>
</tr>
<tr>
<td>Chart</td>
<td>Creates a page with a chart to represent the result of a SQL query.</td>
<td>Available</td>
<td></td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Creating Charts&quot; on page 12-22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column Toggle</td>
<td>Creates a responsive report for mobile applications. By default, column</td>
<td>Not</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Report</td>
<td>toggle reports are created with all columns set to the same priority.</td>
<td>available</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>However, the developer can edit the report column attributes and rank</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>columns by importance. Columns with a lesser priority display at</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>narrower widths (for example, columns with priority 6 will be hidden</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>first and columns with priority 1 will be hidden last.) The report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>includes a Columns button which enables end users to select which</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>columns they want to view.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Developing Reports&quot; on page 10-1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table 7–1 (Cont.) Create Application Wizard - Available Page Types

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>Creates a form to update a single row in a database table. You choose a table on which to build the form.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>List View</td>
<td>Creates a responsive report for mobile applications. The reflow table 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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>List View and Form</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Master Detail</td>
<td>Creates a form that displays a master row and multiple detail rows within a single HTML form. With this form, users can query, insert, update, and delete values from two tables or views. You choose the tables on which to build the master report and detail form.</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Reflow Report</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Report</td>
<td>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 implementation (Interactive or Classic).</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Report and Form</td>
<td>You choose a table on which to build a two page report and form combination. You then choose the report implementation (Interactive or Classic). 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.</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Tabular Form</td>
<td>Creates a form to perform update, insert, and delete operations on multiple rows in a database table. You choose a table on which to build the form.</td>
<td>Available</td>
<td>Not available</td>
</tr>
</tbody>
</table>

**See Also:** "Developing Forms" on page 11-1

"Developing Reports" on page 10-1

"Managing Master Detail Forms" on page 11-12

"Developing Reports" on page 10-1

"Developing Reports" on page 10-1

"Managing Tabular Forms" on page 11-6
Creating a Database Application

This section describes how to create a database application by running the Create Application Wizard.

- Creating a Database Application Based on a Table, Query, or Drill-Down Query
- Creating a Desktop Database Application from a Spreadsheet
- Copying a Database Application
- Deleting an Application

See Also: "Managing User Interface Design" on page 2-8, "Creating Mobile Applications" on page 7-10, "Deleting an Application" on page 7-13, and "What is the Difference Between a Database and Websheet Application?" on page 2-2

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

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 is optimized to display the appropriate page types, attributes, and themes.

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

1. On the Workspace home page, click the Application 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. To learn more, see "Using Themes" on page 13-1 and "Understanding Theme Subscriptions" on page 13-13.
   e. Theme Style - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51. See "Using Theme Styles" on page 13-3.
   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**.

In the Create Application - Add Page dialog:

a. For Select Page Type, select the type of page you want to add. See Table 7–1, "Create Application Wizard - Available Page Types".

   **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.

d. Click **Next**.

**Tip:** To exit this wizard early and use Application Builder Defaults, click the **Create Application** button. To configure Application Builder Defaults, see "Leveraging Application 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. If copying the theme from another application, you are to subscribe the theme. To learn more, see "Understanding Theme Subscriptions" on page 13-13.

7. For Attributes:

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

   - **Application Express Accounts** - Uses the user account credentials created and maintained with the Application Express Service Administration 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.

   - **Database Account** - Requires users logging into your application to enter a database schema name (or user name) and a password in order to authenticate. This account information is managed entirely within the Oracle database.

   - **No Authentication** - Also known as Database Account Credentials. This option enables users to access your application using the account credentials stored in the mod_plsql DAD definition. In most cases this results in users not having to login when accessing your application. This is the quickest way to create a “public” application.

To learn more, see "Establishing User Identity Through Authentication" on page 20-37.
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 the engine determines the application language. The application primary language can be static (that is, derived from the web browser language) or determined from a user preference or item. The database language setting determines date display and sorting characteristics. To learn more, see field-level Help.


d. 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.

e. 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.

f. 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.

g. 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.

h. Click Next.

8. Confirm your selections and click Create Application.

   See Also: "Managing Pages in a Database Application" on page 7-36 and "Developing Reports" on page 10-1
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 Application Builder icon.
2. Click the Create button.
3. Select From a spreadsheet.

4. For Load Method, specify how spreadsheet data is imported. Select one of the following and click Next:
   a. 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 delinate 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 applicable.
      - File Character Set - Choose the character set in which the text file is encoded.
   b. 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 User Interface Defaults:

a. Review the displayed Singular Name and Plural Name.

b. (Optional) Under Column User Interface Defaults, review the displayed column labels.

c. Click Next.

7. For Application Options, specify the following:

a. Application Name - Enter an alphanumeric name for this application.

b. Create Mode:
   
- Read Only does not include insert and update pages.
- Read and Write includes insert and update pages.

c. Report Implementation:
   
- Classic creates a SQL report.
- Interactive creates an interactive report.

d. Click Next.

8. For User Interface Theme, select a theme and click Next.

Themes are collections of templates used to define the layout and style of an entire application. Depending upon the selected theme, Theme Style may also display. See “Using Themes” on page 13-1 and “Using Theme Styles” on page 13-3.

9. Confirm your selections and click Create Application.

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
- Copying an Application from the Application Home Page

Copying a Database Application Using the Create Application Wizard

To copy application by running the Create Application Wizard:

1. On the Workspace home page, click the Application Builder icon.

2. Click the Create button.

3. Select Copy an existing application.
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. Note that Application IDs between 3000 to 9000 are reserved for internal use.
   d. Copy Supporting Object Definitions - Specify whether to include supporting object definitions. Options include:
      − Yes - Include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file.
      − No - Do not include supporting object definitions.
   e. Click Next.

5. Click Create Application.

Copying an Application from the Application Home Page
To copy a database application from the Application home page:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
   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. Note that Application IDs between 3000 to 9000 are reserved for internal use.
   b. New Application Name - Enter a short descriptive name for the application to distinguish the new application in select lists and reports.
c. Copy Supporting Object Definitions - Specify whether to include supporting object definitions. Options include:
   - **Yes** - Include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file.
   - **No** - Do not include supporting object definitions.

d. Click Next.

5. Click **Copy Application**.

---

**Creating Mobile Applications**

Because mobile applications are designed to run specifically on mobile devices, they should be built with minimal, semantic HTML that is optimized for mobile connections. This section describes how to create mobile applications in Oracle Application Express.

- **About jQuery Mobile Support**
- **Creating a Mobile Application**

**See Also:** "Managing User Interface Design" on page 2-8, "About Responsive Design" on page 13-2, and "Deleting an Application" on page 7-13

**About jQuery Mobile Support**

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. To learn more, see:

http://themeroller.jquerymobile.com/

**See Also:** http://jquerymobile.com/ and "Managing User Interface Design" on page 2-8

**Creating a Mobile Application**

To create a mobile application:

1. On the Workspace home page, click the **Application 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. See “Using Themes” on page 13-1.

e. Theme Style - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51. See “Using Theme Styles” on page 13-3.

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 Table 7–1, "Create Application Wizard - Available Page Types".

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 Application Builder Defaults, click the Create Application button. To configure Application Builder Defaults, see “Leveraging Application 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 and "Establishing User Identity Through Authentication" on page 20-37.

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 the engine determines the application language. The application primary language can be static (that is, derived from the web browser language) or determined from a user preference or item. The database language setting determines date display and sorting characteristics. To learn more, see field-level Help.


d. 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.

e. 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.

f. 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.

g. 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.

h. Click Next.

8. Confirm your selections and click Create Application.

See Also: "Managing User Interface Design" on page 2-8
Deleting an Application

You can delete an application from within Application 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
- Deleting an Application from the Edit Application Definition Page

See Also: "Deinstalling Supporting Objects" on page 24-11 and "Removing an Installed Packaged Application" on page 3-5

Deleting an Application from the Application Home Page

To delete an application from Application Builder:
1. On the Workspace home page, click the Application 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.

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 Application 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 the Application Attributes" on page 7-14

Managing Application Attributes

Application attributes apply to an entire application. Once you create an application, you can review and possibly update application attributes.

- Editing the Application Attributes
- Configuring Security Attributes
- Configuring Globalization Attributes
- Managing the Application User Interface
Managing Application Attributes

See Also: "How to Create a Custom Packaged Application" on page 24-6 for information on using the Supporting Objects utility to create a packaged application

Editing the Application Attributes

This section describes how to edit application attributes (also known as the application definition). Common attributes you may want to edit include the application name, application availability, and defined substitution strings.

- Accessing the Edit Application Definition Page
- Edit Application Definition Page

Accessing the Edit Application Definition Page

You edit application attributes on the Edit Application Definition page. To edit the application definition:

1. On the Workspace home page, click the Application 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.

Edit Application Definition Page

The following sections describe regions and attributes on the Edit Application Definition page. In addition to the attributes described here, the Edit Application Definition page displays defined build options, the associated theme, template defaults, and component defaults. Required values are marked with a red asterisk (*).

- Name
- Properties
- Availability
- Error Handling
- Global Notification
- Substitutions
- Build Options
**Name** Use Name attributes to define basic characteristics of your application, including the application name, an optional alphanumeric alias, and a version number. Table 7–2 describes all Name attributes.

### Table 7–2 Application Definition, Name

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Provides a short descriptive name for the application to distinguish it from other applications in your development environment.</td>
</tr>
<tr>
<td>Application Alias</td>
<td>Assigns an alternate alphanumeric application identifier. You can use this identifier for the application ID. For example, suppose you create an alias of myapp for application 105. Using f?p syntax, you could call application 105 as either:</td>
</tr>
<tr>
<td></td>
<td>■ f?p=105:1</td>
</tr>
<tr>
<td></td>
<td>■ f?p=myapp:1</td>
</tr>
<tr>
<td>See Also:</td>
<td>&quot;About Using f?p Syntax to Link Pages&quot; on page 2-17</td>
</tr>
<tr>
<td>Version</td>
<td>Includes the application's version number on a page. You can also automatically tie the version to the date of last modification using the following format masks:</td>
</tr>
<tr>
<td></td>
<td>■ YYYY.MM.DD</td>
</tr>
<tr>
<td></td>
<td>■ MM.DD.YYYY</td>
</tr>
<tr>
<td></td>
<td>■ DD.MM.YYYY</td>
</tr>
<tr>
<td></td>
<td>If your application version uses YYYY.MM.DD, then Application Builder replaces this format mask with the date of last modification of any application attribute.</td>
</tr>
<tr>
<td>Application Group</td>
<td>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.</td>
</tr>
<tr>
<td>See Also:</td>
<td>&quot;Managing Application Groups&quot; on page 7-59</td>
</tr>
</tbody>
</table>

**Properties** Use the Properties section to enable the following attributes: logging, debugging, exact substitutions, application group, feedback, and default error display location. Table 7–3 describes all Name attributes.

### Table 7–3 Application Definition, Properties

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logging</td>
<td>Determines whether user activity is recorded in the Oracle Application Express activity log. 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. This attribute can only be modified if the Application Activity Logging attribute in Oracle Application Express Administration Services is set to Use Application Setting.</td>
</tr>
<tr>
<td>See Also:</td>
<td>&quot;Enabling Application Activity&quot; in Logging in Oracle Application Express Administration Guide.</td>
</tr>
</tbody>
</table>
Managing Application Attributes

Table 7–3 (Cont.) Application Definition, Properties

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debugging</td>
<td>Controls debug mode for the current application. Available options include:</td>
</tr>
<tr>
<td></td>
<td>• Yes - Enables the application to run in a debug mode.</td>
</tr>
<tr>
<td></td>
<td>• No - Prevents end users from running an application in debug mode.</td>
</tr>
<tr>
<td></td>
<td>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 obfuscate application logic.</td>
</tr>
<tr>
<td>Allow Feedback</td>
<td>Enables support for end user feedback for this application. Select Yes or No.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> “Managing Feedback” on page 5-16</td>
</tr>
<tr>
<td>Compatibility Mode</td>
<td>This attribute 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.</td>
</tr>
<tr>
<td>Application Email from</td>
<td>Determines the email address to use as the from address in the application.</td>
</tr>
<tr>
<td>Address</td>
<td>Enter a valid email address to use as the from address when sending email from an email download or subscription. The value can be a literal string containing a valid email or a static substitution reference defined in the application using substitution syntax APP_EMAIL. Examples:</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:john.doe@abc.com">john.doe@abc.com</a></td>
</tr>
<tr>
<td></td>
<td>&amp;MY_APP_EMAIL_FROM.</td>
</tr>
<tr>
<td>Proxy Server</td>
<td>Use this field to specify a proxy server.</td>
</tr>
<tr>
<td></td>
<td>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 Application Builder, you may need to specify a proxy server.</td>
</tr>
<tr>
<td></td>
<td>You can reference values entered into this field from PL/SQL using the PL/SQL package variable APEX_APPLICATION.G_PROXY_SERVER.</td>
</tr>
</tbody>
</table>

**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 7–4.
Table 7–4 Application Definition, Availability

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Specifies whether the application is available or unavailable for use. Options include:</td>
</tr>
<tr>
<td>- Available</td>
<td>Application is available with no restrictions.</td>
</tr>
<tr>
<td>- Available with Developer Toolbar</td>
<td>Application is available for use. For developers, the Developer Toolbar displays on each page. Requires the developer to be logged in to the Application Builder in the same browser session.</td>
</tr>
<tr>
<td>- Available to Developers Only</td>
<td>Application is available to users having developer privileges.</td>
</tr>
<tr>
<td>- Restricted Access</td>
<td>Application is available to developers named in the Restrict to comma separated user list.</td>
</tr>
<tr>
<td>- Unavailable</td>
<td>Application cannot be run or edited. The message in Message for unavailable application displays when users attempt to access the application.</td>
</tr>
<tr>
<td>- Unavailable (Status Shown with PL/SQL)</td>
<td>Application cannot be run or edited. The user is linked to the URL entered in Message for unavailable application.</td>
</tr>
<tr>
<td>- Unavailable (Redirect to URL)</td>
<td>Application cannot be run. The user is linked to the URL entered in Message for unavailable application.</td>
</tr>
</tbody>
</table>


Build Status Identifies the build status of the current application. Options include:
- Run and Build Application - Developers and users can both run and develop the application.

See Also: "Changing Application Build Status Set During Deployment” in Oracle Application Express Administration Guide

Message for unavailable application 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.

Restrict to comma separated user list (status must equal Restricted Access) Use this attribute with the Status Restricted Access. If you set Status to Restricted Access, only the users listed in this attribute can run the application. To use this attribute:
1. From the Status list, select Restricted Access.
2. Enter a comma-delimited list of users who can run the application in the field provided.

Error Handling Use the Error Handling attributes described in Table 7–5 to control or modify how an application logs errors.

Tip: Error handling functions specified here are overridden by similar page-level attributes.
Managing Application Attributes

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" on page 2-25 and "Page Templates" on page 13-42

---

### Table 7-5 Application Definition, Error Handling

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Error Display Location</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>- Inline with Field and in Notification - Error messages display in a notification area defined as part of the page template.</td>
</tr>
<tr>
<td></td>
<td>- Inline with Field - Error messages display within the field label.</td>
</tr>
<tr>
<td></td>
<td>- Inline in Notification - Displays in the #NOTIFICATION_MESSAGE# template substitution string when an error occurs on the page.</td>
</tr>
<tr>
<td>Error Handling Function</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>log_apex_error</td>
</tr>
<tr>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>#OWNER#.log_apex_error</td>
</tr>
<tr>
<td></td>
<td>You must implement error handling functions using the syntax described in the apex_error package.</td>
</tr>
<tr>
<td></td>
<td>function &lt;name of function&gt; ( p_error in apex_error.t_error ) return apex_error.t_error_result</td>
</tr>
<tr>
<td></td>
<td>See Also: apex_error in Oracle Application Express API Reference</td>
</tr>
<tr>
<td></td>
<td>Note: Error handling specified at the page-level overwrites any error handling function specified here.</td>
</tr>
</tbody>
</table>

---

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" on page 2-25 and "Page Templates" on page 13-42
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" on page 2-25

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" on page 24-31

Configuring Security Attributes

You can configure security for your application by configuring attributes on the Edit Security Attributes page. The Security Attributes you choose apply to all pages within an application.

- Accessing the Edit Security Attributes Page
- Security Attributes Page

See Also: "Managing Application Security" on page 20-1

Accessing the Edit Security Attributes Page

To access the Edit Security Attributes page:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. Click Shared Components.
   The Shared Components page appears.
   The Edit Security Attributes page appears.
   The Edit Security Attributes page is divided into sections. When you select a tab 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.

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" on page 7-14.
Security Attributes Page

The following sections describe regions and attributes on the Edit Security Attributes page.

- Authentication
- Authorization
- Session Management
- Session State Protection
- Browser Security
- Database Session

**Authentication**

Authentication is the process of establishing users' identities before they can access an application. Although you can define multiple authentication schemes for your application, only one scheme can be current at a time. Table 7–6 describes the attributes available under Authentication.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public User</td>
<td>Identifies the Oracle schema used to connect to the database through the Database Access Descriptor (DAD). The default value is ANONYMOUS in environments where the database server version is Oracle Database Express Edition and it is APEX_PUBLIC_USER for all other versions of the database server. When APP_USER equals this value, the Application Express engine considers the current session to be a public user session. 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. Note: Previous versions of Oracle Application Express used the built-in substitution string HTMLDB_PUBLIC_USER. 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. By determining if the user is the public user, you can conditionally display or hide information. If the current application user (APP_USER) 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. 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. The Application Express engine also has built in condition types USER_IS_PUBLIC_USER and USER_IS_NOT_PUBLIC_USER.</td>
</tr>
<tr>
<td>Authentication</td>
<td>Select from the authentication schemes defined for the application. To create an authentication scheme, click Define Authentication Schemes.</td>
</tr>
</tbody>
</table>
Managing Application Attributes

Authorization  Authorization controls user access to specific controls or components based on user privileges. Table 7–7 describes the attributes available under Authorization.

Table 7–7  Authorization Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
</table>
| Authorization Scheme       | Make a selection from this list to specify an authorization scheme for your application.  
    For example, if you only want company employees to be able to log in to an application you could define an authorization scheme that is true if the user's email address ends in '@somecompany.com' and then select that authorization scheme for this option. You can assign only one authorization scheme to an entire application. However, you can assign an authorization scheme to individual pages, page controls (such as a region, a button, or an item), or a shared component (such as a menu, a list, or a tab). |
| Define Authorization Schemes | Click Define Authorization Schemes to create an authorization scheme. |
| Run on Public Pages        | Use Run on Public Pages to control whether the application-level authorization scheme is checked on public pages (that is, pages that do not require authorization). Options include:  
    ■ 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.  
    Application-level authorization is never evaluated on the login page (independent of Run on Public Pages), because authorizations typically are dependent on the username. |

See Also:  "Attaching an Authorization Scheme to an Application" on page 20-52

Session Management  Use Session Timeout attributes to reduce exposure to abandoned computers with an open web browser by application. Table 7–8 describes the attributes available under Session Timeout.
### Table 7–8 Session Management

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rejoin Sessions</td>
<td>The Rejoin Sessions attribute 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. A more restrictive instance-level setting overrides this page level value. <strong>Note:</strong> 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” on page 20-5. Rejoin Sessions options include:  - <strong>Disabled</strong> - If the URL does not contain a session ID Application Express creates a new session.  - <strong>Enabled for Public Sessions</strong> - 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.  - <strong>Enabled for All Sessions</strong> - 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 <strong>Unrestricted</strong> or <strong>Checksum Required - Application Level</strong>.  - OR, the URL does not contain payload (a request parameter, clear cache or data value pairs). This option requires that Embed In Frames be set to <strong>Allow from same origin</strong> or <strong>Deny</strong>. This is not tied to a condition about the URL payload, but also applies to session state protected URLs.</td>
</tr>
<tr>
<td>Deep Linking</td>
<td>Use this attribute to enable or prevent deep linking to an application. Options include:  - <strong>Enabled</strong> - The URL to a specific page ultimately redirects there, possibly after the user has logged in.  - <strong>Disabled</strong> - If the URL does not contain a valid session ID, Application Express starts a new session and redirects to the application’s home page. 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. <strong>See Also:</strong> &quot;Security&quot; on page 9-17 to learn more about overriding this behavior at the page-level.</td>
</tr>
<tr>
<td>Maximum Session Length in Seconds</td>
<td>Enter a positive integer to control how many seconds a session exists and should be used by this application. Leave the value null to revert the value to the instance level setting. Enter 0 to have the session exist indefinitely. The session duration may be superseded by the operation of the job that runs every eight hours which deletes sessions older than 12 hours.</td>
</tr>
</tbody>
</table>
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 7-9 describes the attributes available under Session State Protection.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Timeout URL</td>
<td>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. 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 no URL is supplied, the user is redirected to the application home page. Only three substitution items are supported: &amp;APP_SESSION, &amp;SESSION, &amp;APP_ID. Because of the particular purpose of this URL, it is not necessary to include either &amp;APP_SESSION or &amp;SESSION in the link.</td>
</tr>
<tr>
<td>Maximum Session Idle Time in Seconds</td>
<td>Enter a positive integer to control the seconds of inactivity or idle time for sessions used by this application. The idle time is the time between one page request and the next one. Leave the value null to revert the value to the instance level setting. Set to 0 to prevent session idle time checks from being performed.</td>
</tr>
<tr>
<td>Session Idle Timeout URL</td>
<td>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 timeout and to present a login link or other options. If no URL is supplied, the user is redirected to the application home page. Only three substitution items are supported in this URL: &amp;APP_SESSION, &amp;SESSION, &amp;APP_ID. It is not necessary to include either &amp;APP_SESSION or &amp;SESSION in the link.</td>
</tr>
</tbody>
</table>

See Also: "About Utilizing Session Timeout" on page 20-6 and "Configuring Session Timeout" in Oracle Application Express Administration Guide

**Session State Protection** Enabling Session State Protection can prevent hackers from tampering with URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy. Table 7-9 describes the attributes available under Session State Protection.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session State Protection</td>
<td>Make a selection from the Session State Protection list, to enable or disable Session State Protection for your application. Selecting <strong>Enabled</strong> turns on session state protection controls defined at the page and item level.</td>
</tr>
</tbody>
</table>
Table 7–9 (Cont.) Session State Protection

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allows URLs Created After</td>
<td>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 <strong>Expire Bookmarks</strong> button.</td>
</tr>
<tr>
<td>Bookmark Hash Function</td>
<td>Used to create checksums for application level and user level checksums in bookmarkable URLs.</td>
</tr>
<tr>
<td>Expire Bookmarks</td>
<td>Click <strong>Expire Bookmarks</strong> 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. <strong>Tip:</strong> You can also click <strong>Expire Bookmarks</strong> to change the Bookmark Hash Function to switch to a different algorithm for computing checksums.</td>
</tr>
<tr>
<td>Manage Session State Protection</td>
<td>Click <strong>Manage Session State Protection</strong> to configure Session State Protection.</td>
</tr>
</tbody>
</table>

See Also:  "Understanding Session State Protection" on page 20-21

Browser Security  Table 7–10 describes the attributes available under Browser Security.

Table 7–10  Browser Security

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache</td>
<td>Use <strong>Cache</strong> 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. 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. 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 cache-control.</td>
</tr>
</tbody>
</table>
Table 7–11 describes the attributes available under Database Session.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parsing Schema</td>
<td>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).</td>
</tr>
<tr>
<td>Initialization PL/SQL Code</td>
<td>Use this attribute to enter a PL/SQL block that sets a context for the database session associated with the current &quot;show page&quot; or &quot;accept page&quot; 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.</td>
</tr>
<tr>
<td>Cleanup PL/SQL Code</td>
<td>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.</td>
</tr>
</tbody>
</table>
In Application 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.

**Accessing the Globalization Attributes Page**

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
   The Application home page appears.
3. Click Shared Components.
   The Shared Components page appears.
   The Edit Globalization Attributes page appears.

   **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" on page 7-14.

### Configuring Globalization Attributes

In Application 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.

**See Also:** "Providing Security Through Authorization" on page 20-49 and Oracle Label Security Administrator’s Guide

See Also: "Managing Application Globalization" on page 21-1

### Accessing the Globalization Attributes Page

To access the Edit Globalization Attributes page:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
   The Application home page appears.
3. Click Shared Components.
   The Shared Components page appears.
   The Edit Globalization Attributes page appears.

   **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" on page 7-14.

### Edit Globalization Attributes Page

The following sections describe the attributes available on the Edit Globalization Attributes page.

- Application Primary Language
- Application Language Derived From
- Application Date Format
Managing Application Attributes

- Application Date Time Format
- Application Timestamp Format
- Application Timestamp Time Zone Format
- Character Value Comparison
- Character Value Comparison Behavior
- Automatic Time Zone
- Automatic CSV Encoding

**See Also:** "Specifying the Primary Language for an Application" on page 21-4

**Application Primary Language** Identifies the language in which an application is developed. This language is the base language from which all translations are made. For example, suppose application 100 was authored in English, translated into French, and published as application 101. English would be the Application Primary Language.

All modifications to the application should be made to the primary language specified here.

**Application Language Derived From** Determines how Application Builder determines or derives the application language.

The application primary language can be static, derived from the web browser language, or determined from a user preference or item. The database language setting also determines how the date is displayed and how certain information is sorted.

This option enables you to disable browser derived language support. You also have the option of having the application language derived from an application preference. To learn more, see Field-level Help.

**Application Date Format** Determines the date format to be used in the application.

Use this date format to alter the `NLS_DATE_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If no value is specified, the default date format is derived from the database session at runtime. Consider the following examples:

- Month DD, YYYY
- `&MY_DATE_FORMAT`.

**Application Date Time Format** Specify the date time format to be used in the application.

This date time format can be referenced in an application using the substitution reference `&APP_DATE_TIME_FORMAT`, or in PL/SQL using the function `v('APP_DATE_TIME_FORMAT')`. This attribute does not alter any NLS settings. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If this attribute value is not specified, then a reference to `APP_DATE_TIME_FORMAT` returns the NLS database session date format and the NLS time format. Consider the following examples:

- Month DD, RRRR HH24:MI
- `&MY_DATE_FORMAT`.
Managing Application Attributes

**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.
```

**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.
```

**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 the `NLS_SORT` database session parameter for the execution of SQL queries in classic report and interactive report regions. If no value is specified, the default value is derived from the database session at runtime. Consider the following examples:

```
BINARY
GERMAN
CANADIAN_M
```

**Character Value Comparison Behavior** Determines the collation behavior of SQL operations, for example, `LIKE`, `MIN/MAX`.

Use this value to alter the `NLS_COMP` database session parameter for the execution of SQL queries in classic report and interactive report regions. Options include:

- **Database session NLS setting (default)** - The `NLS_COMP` value is derived from the database session at runtime.
- **Binary** - Comparisons in `WHERE` clauses and other SQL operations are binary.
- **Linguistic** - Comparisons in `WHERE` clauses and other SQL operations use the linguistic sort specified in the Character Value Comparison attribute (`NLS_SORT`).

**Automatic Time Zone** Controls the setting of the database session time zone. When set to `Yes`, the client time zone is derived from the client’s web browser and set for the duration of the Application Express session.

Subsequent page views have the database session time zone set properly per page view. Once set, this setting can be overridden using `APEX_UTIL.SET_SESSION_TIME_ZONE`, or reset using `APEX_UTIL.RESET_SESSION_TIME_ZONE`. 
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 Application Builder is determined by the character set of the Database Access Descriptor (DAD) used to access Oracle Application Express. For example, if the character set of the Database Access Descriptor is AL32UTF8, all pages in all applications in the Oracle Application Express user interface are encoded in UTF-8.

By default, the CSV output from report regions is encoded in the same character set as the Database Access Descriptor. However, some desktop spreadsheet applications require that the data is encoded in the client desktop operating system character set. In the case of multibyte data, the CSV output from report regions often appears corrupted when opened by a desktop spreadsheet application. This is because the CSV output from report regions is encoded differently than what is required by the desktop application. Enabling Automatic CSV Encoding resolves this issue.

For example, if the user’s language preference for an application is de, the CSV data is encoded in Western European Windows 1252, regardless of the Database Access Descriptor character set setting. If the user’s language preference is zh-cn, the CSV data is encoded in Chinese GBK.

Managing the Application User Interface

When you create a database application you select a user interface. The application user interface (such as Desktop or Mobile) determines default characteristics of the application and optimizes the display for the target environment. You can customize the user interface by editing attributes on the User Interface and User Interface Details page.

- Accessing User Interface Attributes
- User Interface Page
- Defining an Application Logo
- Creating a New User Interface
- User Interface Details Page

Accessing User Interface Attributes

To access user interface attributes:

1. On the Workspace home page, click the Application 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 "User Interface Page" on page 7-30.

5. To edit attributes for a specific user interface, scroll down to User Interfaces and select a specific user interface.

The User Interface Details page appears. To learn more, see "User Interface Details Page" on page 7-32.

**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" on page 7-14.

### User Interface Page

This section describe the sections and attributes available on the User Interface page.

- General Properties
- Logo
- User Interface Detection
- User Interfaces

**See Also:** "Accessing User Interface Attributes" on page 7-29

**Tip:** To learn more about the attributes on this page, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

### General Properties

Use General Properties to define basic characteristics of the application user interface. Table 7–12 describes all General Properties attributes.

#### Table 7–12 User Interface, General Properties

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| Static File Prefix         | Determines the virtual path the web server uses to point to the static files of the application. Use this attribute to centrally change where static application files should be loaded from without changing all file references in an application. For example, you can use this attribute to point to a directory on the web server to avoid having to load files from the database and improve application performance. If you specify a Static File Prefix, the substitution string `#APP_IMAGES#` returns that value. For example, you could enter a valid URL to reference them: `/myFiles/` or `http://contentDeliveryNetwork.com/myFiles/`
|                            | If you do not specify a Static File Prefix, the substitution string `#APP_IMAGES#` returns a URL to reference files from Shared Components, Static Files which will load the files from the database. **Tip:** Do not use this attribute to reference files which are stored with your application definition in the database. |
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" on page 7-31

**User Interface Detection** Enter CSS file URLs for style sheets that should load when Oracle Application Express displays the available user interfaces, when automatic detection fails. Each URL has to be written on 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 also have access to the substitution string #APP_VERSION# to include the application's version in the file URL. If you do not enter any URLs, Oracle Application Express uses a default style sheet.

To view examples, see field-level Help.

**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" on page 7-29, "Creating a New User Interface" on page 7-32, and "User Interface Details Page" on page 7-32

**Defining an Application Logo**

To define an application logo:

1. Access the User Interface page. See "Accessing User Interface Attributes" on page 7-29.

   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**.

   **See Also:** "Managing Static Application Files" on page 18-8, "Managing Static Workspace Files" on page 18-12, "Creating Custom Themes" on page 13-23, and "Page Templates" on page 13-42

---

**Creating a New User Interface**

To add a new user interface:

1. Navigate to the User Interface page. See "Accessing User Interface Attributes" on page 7-29.
2. Scroll down to User Interfaces and click **Add New User Interface**.
3. Follow the on-screen instructions.
   
   To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
4. Confirm your selections and click **Create**.

   **See Also:** "Accessing User Interface Attributes" on page 7-29 and "User Interface Details Page" on page 7-32

---

**User Interface Details Page**

This section describes the sections and attributes available on the User Interface Details page.

- **Identification**
- **Attributes**
- **Navigation Bar**
- **JavaScript**
- **Cascading Style Sheets**
Concatenated Files

See Also: "Accessing User Interface Attributes" on page 7-29 to learn how to access the User Interface Details page.

Tip: To learn more about the attributes on this page, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

Identification Table 7–13 describes all attributes under User Interface Details, Identification.

Table 7–13 User Interface Details, Identification

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface Type</td>
<td>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.</td>
</tr>
<tr>
<td>Display Name</td>
<td>Specify a display name for the user interface. The display names is shown in wizards, such as the create theme wizard.</td>
</tr>
<tr>
<td>Sequence</td>
<td>Specify the display sequence for the user interface.</td>
</tr>
</tbody>
</table>

Attributes Table 7–14 describes all attributes under User Interface Details, Attributes.

Table 7–14 User Interface Details, Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Detect</td>
<td>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.</td>
</tr>
<tr>
<td>Default</td>
<td>Select whether the user interface is the default interface for the application.</td>
</tr>
<tr>
<td>Home URL</td>
<td>Specify the home page of the application for the current user interface.</td>
</tr>
<tr>
<td>Login URL</td>
<td>Specify the login page of the application for the current user interface.</td>
</tr>
<tr>
<td>Theme</td>
<td>Shows the theme currently associated with the user interface. See Also: &quot;Switching Themes&quot; on page 13-16</td>
</tr>
<tr>
<td>Theme Style</td>
<td>Select a theme style. This option only displays for newer themes that support theme styles.</td>
</tr>
<tr>
<td>Global Page</td>
<td>If defined, displays the global page for the application. See Also: &quot;Creating a Global Page to Display Components on Every Page&quot; on page 14-3</td>
</tr>
</tbody>
</table>

Navigation Bar Navigation Menu attributes only display with newer themes. Table 7–15 describes Navigation Menu attributes.
JavaScript  Use the attributes described in Table 7–16 to control or modify how an application handles JavaScript.

Table 7–16  User Interface Details, JavaScript

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Delivery Network</td>
<td>Specify the Content Delivery Network used to load the 3rd party libraries jQuery and jQuery Mobile. Using a Content Delivery Network can reduce the loading time of your application if the user has visited other websites which also use the same content delivery network to load the jQuery files. It is very likely that those calls are still in the browser cache and do not have to be downloaded again. Especially on mobile devices with limited browser caches, this can improve performance because it is not so likely that the files are purged from the cache. <strong>Note:</strong> The Google and Microsoft Content Delivery Network do not currently host the correct version of the jQuery Mobile library, so selecting either of these will only influence the jQuery library being served from their Content Delivery Network, jQuery Mobile will still be served from your web server.</td>
</tr>
</tbody>
</table>
Table 7–16 (Cont.) User Interface Details, JavaScript

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| File URLs          | Enter JavaScript file URLs for code to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file, you can use the substitution string #MIN# to include .min or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. JavaScript file URLs you enter here replaces the #APPLICATION_JAVASCRIPT# substitution string in the page template. **Note:** You do not need to include opening or closing script tags. Just write the URL. Examples:  
  - Standard file reference:  
    /myjs/main.js  
  - Standard file reference which loads the minified file main.min.js for regular page views and main.js in Debug mode:  
    /myjs/main#MIN#.js  
  - Conditional file for Internet Explorer  
    [if IE]/myjs/ie.js |
| Include Legacy JavaScript | 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. 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. **See Also:** "Legacy JavaScript APIs" in Oracle Application Express API Reference |
| Include jQuery Migrate | Specifies if the jQuery Migrate plug-in should be included on every page in the application. 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. 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. |

**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. See field-level Help for examples.
**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.

---

**Managing Pages in a Database Application**

A page is the basic building block of an application. When you build an application in Application Builder, you create pages that contain user interface elements, such as tabs, lists, buttons, items, and regions.

- **About Adding Pages to a Database Application**
- **Available Pages Types When Running the Create Page Wizard**
- **Creating a Page from the Application Home Page**
- **Creating Dialog Pages**
- **About Managing Pages in Page Designer**
- **About Managing Pages in Component View**
- **About Copying Database Application Pages**
- **About Copying Pages Between User Interfaces**
- **About Deleting Database Application Pages**

See Also: "Accessing Application Utilities" on page 7-61

---

**About Adding Pages to a Database Application**

You add a page to an existing page in a database application by running the Create Page Wizard. You can access this wizard by:

- On the Application home page, click **Create Page**. See "Creating a Page from the Application Home Page" on page 7-40.
- When viewing a page in Component View, click **Create** and select **Page**. See "Managing Pages in Component View" on page 9-1.
- When viewing a page in Page Designer, click **Create** and select **Page**. See "Managing Pages in Page Designer" on page 8-1.

---

**Available Pages Types When Running the Create Page Wizard**

Table 7–17, "Available Pages Types When Running the Create Page Wizard" lists the types of pages you can create when running the Create Page Wizard. The supported page types differ depending on the selected user interface (for example, Desktop and jQuery Mobile Smartphone).
<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Control</td>
<td>Creates a page containing an access control list, enabling developers to control access to an application, individual pages, or page components.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>See Also:</strong></td>
<td>&quot;Controlling Access to Applications, Pages, and Page Components&quot; on page 20-32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blank Page</td>
<td>Creates a page with no built-in functionality.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Calendar</td>
<td>Generates a calendar with monthly, weekly, and daily views.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>See Also:</strong></td>
<td>&quot;Creating Calendars&quot; on page 12-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chart</td>
<td>Enables you to create HTML5 and Flash graphical charts.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>See Also:</strong></td>
<td>&quot;Creating Charts&quot; on page 12-22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Loading</td>
<td>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.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td><strong>See Also:</strong></td>
<td>&quot;Creating Applications with Data Loading Capability&quot; on page 25-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback Page</td>
<td>Adds a feedback page. Feedback is the process of gathering real-time comments, enhancement requests, and bugs from your application users.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>See Also:</strong></td>
<td>&quot;Managing Feedback&quot; on page 5-16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form, Form and Report on Web Service</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>See Also:</strong></td>
<td>&quot;Creating an Input Form and Report on a Web Service&quot; on page 25-15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form, Form on a Procedure</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form, Form on a SQL Query</td>
<td>Creates a form based on the columns returned by a SQL query such as an EQUJOIN.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>Form, Form on a Table or View</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form, Form on a Table with List View</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Desktop applications do not support this report format.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form, Form on a Table with Report</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>This wizard does not support tables having more than 127 columns. Selecting more than 127 columns generates an error.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 7–17 (Cont.) Available Pages Types When Running the Create Page Wizard

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form, Form on Web Service</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating a Form on a Web Service&quot; on page 25-17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form, Master Detail Form</td>
<td>Builds a two page report and form combination. On the first page, users select a row to update. On the second page, users can update the selected table or view and its associated detail. You choose the tables on which to build the master report and detail form.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Managing Master Detail Forms&quot; on page 11-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form, Summary Page</td>
<td>Creates a read-only version of a form. Typically used to provide a confirmation page at the end of a wizard.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form, Tabular Form</td>
<td>Creates a form to perform update, insert, and delete operations on multiple rows in a database table. You choose a table on which to build the form.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Managing Tabular Forms&quot; on page 11-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global Page</td>
<td>A global page (previously referred to as Page 0) functions as a master page.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating a Global Page to Display Components on Every Page&quot; on page 14-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Login Page</td>
<td>Creates a login page.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating a Login Page&quot; on page 20-48</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Map Chart</td>
<td>Creates a Flash map based on the AnyChart AnyMap Interactive Maps Component.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating Maps&quot; on page 12-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plug-ins</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Importing a Plug-in from the Plug-in Page&quot; on page 19-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report, Classic Report</td>
<td>Formatted result of a SQL query. You can 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. For assistance in writing the SQL query, click the Query Builder button.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Understanding the Difference Between Interactive and Classic Reports&quot; on page 10-1 and &quot;Editing Classic Reports in Component View&quot; on page 10-72</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 7–17 (Cont.) Available Pages Types When Running the Create Page Wizard

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report, Column Toggle Report</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Report, Interactive Report</td>
<td>Formatted result of a SQL query. Creates an interactive report based on a custom SQL SELECT statement you provide. For assistance in writing the SQL query, click the Query Builder button. Interactive reports are only supported on Desktop pages and you can only define one interactive report per page. End users can customize the report layout and data displayed by selecting options on the Actions menu. Note: jQuery Mobile Smartphone does not support interactive reports. See Also: &quot;Understanding the Difference Between Interactive and Classic Reports&quot; on page 10-1, &quot;Customizing Interactive Reports as an End User&quot; on page 10-57, and &quot;Editing Interactive Reports in Component View&quot; on page 10-33</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Report, List View</td>
<td>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. Note: Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Report, Reflow Report</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Report, Report on Web Service</td>
<td>Creates a report on a Web Service result. See Also: &quot;Implementing Web Services&quot; on page 25-7</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Tree</td>
<td>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 Also: &quot;Creating Trees&quot; on page 15-49</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Wizard</td>
<td>Create a wizard.</td>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>
Creating a Page from the Application Home Page

To create a page from the Application home page:

1. On the Workspace home page, click Application 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 Table 7–17, "Available Pages Types When Running the Create Page Wizard".
   c. Click Next.
5. Follow the on-screen instructions.

See Also: "About Adding Pages to a Database Application" on page 7-36, "Editing Page Attributes in Page Designer" on page 8-19, and "Configuring Dialog Attributes in Component View" on page 9-23

Creating Dialog Pages

This section describes how to create dialog pages.

- About Supported Dialogs
- Using a Dialog Page Template
- Defining a Page as a Dialog in Component View
- Defining a Page as a Dialog in Page Designer
- Generating a Cancel Dialog Dynamic Action on the Dialog Page
- Generating a Close Dialog Page Process on the Dialog Page
- About Branching with Dialog Pages
- Handling Refresh Actions on Parent Page

About Supported Dialogs

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. 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 (that is, a modal dialog, or a non-modal dialog). Oracle Application Express supports two types of dialog pages:

- **Modal Dialog** - A modal dialog is an overlay window positioned within the same browser window. A modal dialog remains active and focused until the user has finished with it and closes it. While a modal dialog is active, the user is unable to interact with the rest of the page until the dialog is closed.

- **Non-modal Dialog** - A non-modal dialog displays a separate popup browser window. A user can interact with a non-modal dialog and content on the page. Use a non-modal dialog when the requested information is not essential to continue. This type of window can be left open while work continues elsewhere.
Using a Dialog Page Template
Oracle Application Express includes dialog page templates in the *Universal Theme - 42*. When you set the Page Mode to either *Modal* or *Non-Modal*, Oracle Application Express automatically uses the default dialog page template. This section describes how to create a dialog template if your theme does not include one and how to edit dialog template attributes.

Creating a Dialog Page Template 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 *Application 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" on page 13-25.

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. To learn more about configuring these template attributes, see "Required Dialog Page Template Attributes" on page 13-48.

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:

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.


Defining a Page as a Dialog in Page Designer
To define a page as a dialog in Page Designer:


See Also: "About Adding Pages to a Database Application" on page 7-36

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" on page 15-30

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" on page 17-17, "Understanding Sample Database Application" on page 3-6, and "Controlling Navigation Using Branches" on page 16-20

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 the parent (or launching) page or another Normal page in an application.

### Handling Refresh Actions on Parent Page

If you have generated a Close Dialog page process on the dialog page, you must implement refresh actions on the parent page using the new *Dialog Closed* dynamic action event type. You can use the set type *Dialog Return Item* when creating a *Set Value True Action*, to set a single value, based on the return item of a dialog.

Consider the following example. Edit the Products in the *Sample Database Application*. (See "Products Page" on page 3-10) On the Products page, the Create Product button has an associated dynamic action that launches the dialog. Closing of 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:** "Managing Dynamic Actions" on page 15-30 and "Understanding Sample Database Application" on page 3-6

### About Managing Pages in Page Designer

Page Designer is a full featured Integrated Development Environment (IDE) designed to maximize developer productivity in maintaining application pages. The Page Designer window is divided into three main sections.

- **Tree Pane** - The Tree pane displays in the left column and features the following tree views: Rendering, Dynamic Actions, Processing, and Shared Components.

- **Grid Layout** - Grid Layout displays in the center column. Grid Layout is a visual representation of the regions, items, and buttons that define a page. You can edit the position of page components by dragging and dropping them. You can also add new regions, items, and button by dragging and dropping them from Gallery.

- **Property Editor** - The Property Editor displays in the right column. Use the Property Editor to edit the properties of the component currently selected in the Tree Pane or Grid Layout.

**Tip:** The Rendering tree and Grid Layout are two different representations of the regions, items, and buttons that define a page.

**See Also:** "Managing Pages in Page Designer" on page 8-1
About Managing Pages in Component View

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.

The 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 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: "Managing Pages in Component View" on page 9-1

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" on page 8-25 and "Copying a Database Application Page in Component View" on page 9-26

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
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" on page 8-26 and "Deleting Pages in Component View" on page 9-26

Using the Find Icon
This section describes how to use the Find icon in Component View.

- Using the Find Icon
- Searching for Items
- Searching for Pages
- Searching for Queries
- Searching for Tables
- Using the PL/SQL Finder
- Searching for Images

About the Find Icon
The Find icon resembles a flashlight and displays on many pages in Application Builder including Application home page, 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
To use the Find icon.
1. Click the Find icon.

The Items Finder appears.
2. Click one of the following tabs:
Searching for Items

In Application 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.

See Also:  "Understanding Page-Level Items" on page 15-1

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. See "About the Actions Menu" on page 10-58.

2. To edit a specific item, navigate to the appropriate item. See "Understanding Page-Level Items" on page 15-1.

### 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. See "About the Actions Menu" on page 10-58.

3. To link to the Items page, click the page number.

### Searching for Queries

You can use the Queries Finder to locate a query within your application or within the schema associated with the workspace.

To search for a query using the Queries Finder:

1. Click the **Find** icon.
2. Click the **Queries** tab.

The Queries Finder appears. A search bar displays at the top of the page and contains the following controls:

- **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search.
- **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report. See "About the Actions Menu" on page 10-58.

3. To link to the Items page, click the page number.

### Searching for Tables

You can use the Tables Finder to view tables within the schema associated with the workspace.

To view tables associated within the current schema:

1. Click the **Find** icon.

2. Click the **Tables** tab.

The Tables Finder appears. A search bar displays at the top of the page and contains the following controls:

- **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search.
- **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report. See "About the Actions Menu" on page 10-58.

3. Select a table name.

A table definition appears.
Using the Find Icon

This report displays the column names, data type, length, precision, and scale and the SQL necessary to re-create the table appears at the bottom of the page.

See Also: "Managing Tables" in Oracle Application Express SQL Workshop Guide

Using the PL/SQL Finder

You can use the PL/SQL Finder to locate and view details about stored procedures, functions, and packages associated with each object within the schema associated with the workspace.

To search for PL/SQL code in the current schema:

1. Click the Find icon.

2. Click the PL/SQL tab.

   The PL/SQL Finder appears. A search bar displays at the top of the page and contains the following controls:

   - **Select columns to search icon** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select All Columns.
   - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click Go.
   - **Go button** - Executes a search.
   - **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize an interactive report. See "About the Actions Menu" on page 10-58.

3. To view additional details, select the procedure, function, or package name.

   The procedure, package, or function name appears and additional information including the owner, source name, source type, return type, argument names, data types, and IN/OUT parameters.

   See Also: "Managing Database Objects with Object Browser" in Oracle Application Express SQL Workshop Guide

Searching for Images

You can use the Images Finder to identify images available to the current application.

To view available images:

1. Click the Find icon.
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

The section describes different ways to view page groups.

To access the Page Groups page:

1. On the Workspace home page, click the Application 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.
Creating Page Groups

- **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" on page 10-58

Creating a Page Group

To create a page group:

1. On the Workspace home page, click the **Application Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. On the Page Groups page, click **Create**.
6. Enter a name, a description (optional), and click **Create**.

Editing a Page Group Definition

When you create a page group you specify a name and description.

To edit the Page Group definition:

1. On the Workspace home page, click the **Application Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. Select a page group.
6. On the Page Group page, edit the name or description and click **Apply Changes**.
7. To move to the next Page Group definition, click the **Previous** (<) and **Next** (>) buttons and repeat the previous step.

Assigning a Page to a Page Group

To assign pages to a group:

1. On the Workspace home page, click the **Application 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.

See Also: You can also associate a page with a group by editing the page attributes and selecting the group, or when running the Create Page Wizard. See "Editing Page Attributes in Page Designer" on page 8-19, "Editing Page Attributes in Component View" on page 9-11, and "Using the Create Application Wizard" on page 7-1.

**Viewing Pages by Page Group**

To view pages by page group:

1. On the Workspace home page, click the Application 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. Clicking a page number takes you to the Page Attributes page. Clicking the page Name links to the page.

6. To create a new group, click Create and follow the on-screen instructions.

**Deleting a Page Group**

To delete a page group:

1. Remove all pages from the group to be deleted. See "Viewing Pages by Page Group" on page 7-52.

2. Access the Page Group page:

   a. On the Workspace home page, click the Application Builder icon.

   b. Select the application.

   c. Click the Utilities icon.

   d. From Page Specific Utilities, click Page Groups.

3. Select a page group.

4. Click Delete.

   A confirmation page appears.

5. Confirm your request.

**Locking and Unlocking a Page**

You can prevent conflicts during application development by locking pages in your application. By locking a page, you prevent other developers from editing it.

- Determining If a Page Is Locked
Locking and Unlocking a Page

- Locking a Page
- Unlocking a Page

Determining If a Page Is Locked

A lock icon indicates whether a page is currently locked. If a page is unlocked, the icon appears as an open padlock. If the page is locked, the icon appears as a locked padlock. A lock icon appears on the Application home page and on the page.

To view the lock icon on the Application home page, click the View Report icon. A report appears. The lock icon appears on the right side of the page.

<table>
<thead>
<tr>
<th>Page</th>
<th>Name</th>
<th>Updated</th>
<th>Updated By</th>
<th>Page Type</th>
<th>Group</th>
<th>User Interface</th>
<th>Lock</th>
<th>Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Home</td>
<td>3 weeks ago</td>
<td>-</td>
<td>Home</td>
<td>Unassigned</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Interactive Report</td>
<td>67 seconds ago</td>
<td>admin_user</td>
<td>Interactive Report</td>
<td>Unassigned</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Access Control Administration Page</td>
<td>23 hours ago</td>
<td>admin_user</td>
<td>DML Form</td>
<td>Unassigned</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>Login Page</td>
<td>3 weeks ago</td>
<td>-</td>
<td>Login</td>
<td>Unassigned</td>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Locking a Page

This section describes the different ways you can lock a page.

- Locking a Page from the Application Home Page
- Locking a Page from the Page Locks Page
- Locking a Page in Page Designer
- Locking a Page in Component View

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 Application Builder icon.
2. Select an application.
3. On the search bar, click the View Report or View Detail icon.
4. Click the Lock icon.
5. Enter a comment in the Comment field.
6. Click Lock Page(s).

Locking a Page from the Page Locks Page

To lock a page from the Page Locks Page:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. Click the Utilities icon.
4. On the Page Specific Utilities list, click **Cross Page Utilities**.
5. Click **Page Locks**.
6. Select the appropriate pages and click **Lock Checked**.
7. Enter a comment in the Comment field.
8. Click **Lock Page(s)**.

**Locking a Page in Page Designer**
The Page Lock button on the Page Designer toolbar indicates whether a page is locked. If the page is unlocked, the Page Lock button displays as an unlocked padlock.

To lock a page in Page Designer:
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.

**Locking a Page in Component View**
To lock a page in Component View:
1. Navigate to the appropriate page in Component View. See "Viewing a Page in Component View" on page 9-5.
2. Click **Utilities** and then **Lock**.
3. Enter a comment in the Comment field.
4. Click **Lock Page(s)**.

**Unlocking a Page**
This section explains how to unlock a locked page. Only the developer who locks a page can unlock it. However, a developer with administrative privileges can unlock pages locked by other developers.

- **Unlocking Pages from Utilities**
- **Unlocking Pages from the Application Home Page**
- **Unlocking a Page in Page Designer**
- **Unlocking a Page in Component View**
- **Unlocking Pages As a Workspace Administrator**

**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**
To unlock a page from the Page Locks page:
1. On the Workspace home page, click the **Application Builder** icon.
2. Select an application.
3. Click the Utilities icon.
4. On the Page Specific Utilities list, click Cross Page Utilities.
5. Click Page Locks.
6. Select the appropriate pages.
7. Click Unlock Checked.

Unlocking Pages from the Application Home Page
To unlock a page from the Application Home Page:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the search bar, click the View Report icon.
4. Click the Lock icon.
   The Lock Details page appears.
5. Click Unlock.

Unlocking a Page in Page Designer
The Page Lock button on the Page Designer toolbar indicates whether a page is locked. If the page is locked, the Page Lock button displays as a locked padlock. Passing your cursor over a locked Page Lock button displays a tooltip which indicated who locked it.

To unlock a page in Page Designer:
1. Navigate to the appropriate page. See "Viewing a Page in Page Designer" on page 8-15.
   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.

Unlocking a Page in Component View
To unlock pages in Component View:
1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
2. Click Utilities and select Unlock.

Unlocking Pages As a Workspace Administrator
To unlock pages as a Workspace Administrator:
1. On the Workspace home page, click the Application 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).

Adding Developer Comments

You can add developer comments to an application, a page, or a group of pages using the Developer Comment, Bug, or To Do button. You can use developer comments to communicate application changes, report issues, or record developer suggestions.

- Developer Comment, Bug, or To Do Button
- Adding Developer Comments to an Application
- Viewing and Editing Developer Comments
- Deleting Developer Comments
- Accessing the Developer Comments from Shared Components
- Viewing the Developer Comments Calendar

See Also: "Adding Database Application Comments" on page 7-58

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 Application Builder that relate to a specific application or application page.

Adding Developer Comments to an Application

You can add developer comments to an application, a page, or a group of pages.

To add a developer comment:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. Click the Developer Comment, Bug, or To Do button.
4. From the Type list, select Developer Comment.
   The Developer Comments page appears.
5. In Comment, enter up to 4000 characters of text.
6. Click Create.

Viewing and Editing Developer Comments

To edit a developer comment:
1. On the Workspace home page, click the Application 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” on page 7-58.

**Deleting Developer Comments**

This section describes how to delete developer comments.

- Deleting a Specific Developer Comment
- Deleting Multiple Developer Comments

**Deleting a Specific Developer Comment**

To delete a developer comment:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. Click the Developer Comment, Bug, or To Do icon.
4. Click View Comments.
   The Developer Comments page appears.
5. Locate the comment to be deleted.
6. Click the Edit icon.
   The Edit Comment page appears.
7. Click Delete.

**Deleting Multiple Developer Comments**

To delete multiple developer comments:
1. On the Workspace home page, click the Application 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:
Adding Database Application Comments

- Delete all comments
- Delete comments created by a developer
- Delete comments by date

7. Follow the on-screen instructions.

Accessing the Developer Comments from Shared Components

To access the Developer Comments report from Shared Components:

1. On the Workspace home page, click the Application 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" on page 10-58.

See Also: "Viewing and Editing Developer Comments" on page 7-56 and "Deleting Developer Comments" on page 7-57

Viewing the Developer Comments Calendar

To access the Developer Comments Calendar:

1. On the Workspace home page, click the Application 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.

Adding Database Application Comments

You 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.
Managing Application Groups

To create an application comment:

1. Navigate to the Workspace home page.
2. Click the Application Builder icon.
3. Select an application.
4. On the Application home page, click Shared Components. 
The Shared Components page appears.
5. Under Tasks, click Edit Application Comment. 
The Application Comments page appears.
6. Enter comments in the Comments field and click Apply Changes.

Managing Application Groups

You can organize your applications by assigning them to application groups. To use application groups, first create a group and then assign applications to it.

- Viewing Database Application Groups
- Creating a Database Application Group
- Assigning a Database Application to an Application Group
- Removing a Database Application from an Application Group
- Deleting a Database Application Group

See Also: "Utilizing Cross Page Utilities" on page 7-66

Viewing Database Application Groups

To view database application groups:

1. On the Workspace home page, click the Application Builder icon.
2. Click Workspace Utilities.
3. Click Application Groups.

The Application Group page appears.

A Search bar displays at the top of the page. Available controls include:

- Select columns to search - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select All Columns.
- Text area - Enter case insensitive search criteria (wildcard characters are implied) to search for a group by name and click Go.
- Go button - Executes a search or applies a filter.
- View Icons (the default) displays each group as a large icon. To edit a group, click the appropriate icon.
- View Report displays each group as a line in a report. To edit a group, click the name.

4. To view the applications associated with a group, you click the group name.
Creating a Database Application Group

To create a database application group:
1. On the Workspace home page, click the Application Builder icon.
2. Click Workspace Utilities.
3. Click Application Groups.
4. On the Application Group page, click Create.
5. Enter a name, a description (optional), and click Create.

Assigning a Database Application to an Application Group

To assign a database application to a group:
1. On the Workspace home page, click the Application 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 the Application Attributes" on page 7-14.

Removing a Database Application from an Application Group

To remove an application from an application group:
1. On the Workspace home page, click the Application 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 the Application Attributes" on page 7-14.

Deleting an Database Application Group

To delete an application group:
1. On the Workspace home page, click the Application Builder icon.
2. Click Workspace Utilities.
3. Click **Application Groups**.

4. Assign all applications in the group to Unassign or assign them to other groups. See “Removing a Database Application from an Application Group” on page 7-60.

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.

### Accessing Application Utilities

This section describes Application Builder utilities and reports that enable you to view your application from various perspectives. You can use these reports to view statistics and access reports of page and application components, templates, plug-ins, and conditions. For example, you can view details about buttons used on all pages within your application. Additionally, many reports are updatable so you can standardize components, such as item and region labels, without navigating to a specific page.

- **Using Application Builder Utilities**
- **Accessing Page Specific Utilities**
- **Utilizing Cross Page Utilities**
- **Accessing Cross Application Reports**
- **Using Interactive Report Utilities**

**See Also:** "Creating Custom Activity Reports Using APEX_ACTIVITY_LOG" on page 23-6

### Using Application Builder Utilities

This section describes the tools and reports available on the Application Builder Utilities page.

- **Accessing the Utilities Page**
- **Viewing the Application Dashboard**
- **Viewing the Change History Report**
- **Viewing the Recently Updated Pages Report**
- **Upgrading Applications to Include New Components**
- **Using the Database Object Dependencies Report**
- **Accessing Application Express Views**

### Accessing the Utilities Page

To access Application Builder Utilities page:

1. On the Workspace home page, click the **Application Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.

The Utilities page features icons that link to the following tools and reports:
Accessing Application Utilities

- **Application Dashboard.** See "Viewing the Application Dashboard" on page 7-62.
- **Change History.** See "Viewing the Change History Report" on page 7-63.
- **Recently Updated Pages.** See "Viewing the Recently Updated Pages Report" on page 7-63.
- **Debug Messages.** See "Debugging an Application" on page 22-1.
- **Advisor.** See "Running Advisor to Check Application Integrity" on page 22-1.
- **Upgrade Application.** See "Running Upgrade Application" on page 7-64.
- **Database Object Dependencies.** See "Using the Database Object Dependencies Report" on page 7-64.
- **Attribute Dictionary.** - See "Using the Attribute Dictionary" on page 17-32.
- **Application Express Views.** - See "Accessing Application Express Views" on page 7-65.

**See Also:** "Accessing Page Specific Utilities" on page 7-65 and "Utilizing Cross Page Utilities" on page 7-66

**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 **Application 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" on page 20-15

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 Application Builder icon.
3. Select an application.
   The Application home page appears.
4. Click Utilities.
5. Click Change History.
   A Search bar displays at the top of the page. See "About the Search Bar" on page 10-57.

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 Application 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. See "About the Search Bar" on page 10-57.

Upgrading Applications to Include New Components
This section describes how to upgrade an existing application to include new components available in the current release.

About Upgrading an Existing Application
Running Upgrade Application

About Upgrading an Existing Application Use the Upgrade Application icon on the Utilities page upgrades the following components:

- Update Text Field items to Number Field items when there is an IS NUMERIC validation.
- Update the Value Required item attribute to **Yes** when there is a NOT NULL validation.
- Upgrade Date Picker (Classic) to the new Date Picker.
- Remove `%null%` in LOV Null Return Value of page items.
- Upgrade FCKeditor 2 to CKEditor 4.
- Upgrade SVG Charts to HTML5 Charts.
- Upgrade HTML Charts to HTML5 Charts.
- Upgrade Flash Charts to HTML5 Charts.
- For interactive reports:
  - Enable Group By for interactive reports.
  - Enable Save Public Report for interactive reports.
  - Enable Subscription for interactive reports.
  - Enable Rows Per Page Selector for interactive reports.
  - Enable fixed report headers to the top of the page.
- Numeric, Required, and Date Picker Item updates based upon conditional validations.

  A wizard displays all relevant validations per page and enables you to select the ones you want to replace with item settings. Then, select buttons that should not execute validations (such as Cancel). Lastly, you can select any validations that should always execute (for example, checking privileges before deletion).

  **See Also:** "Running Upgrade Application" on page 7-64

### Running Upgrade Application

To upgrade an existing application:

1. Navigate to the Workspace home page.
2. Click the **Application 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 an Existing Application" on page 7-63

### 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 **Application 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.

**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 Application 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" on page 10-58.

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.

**Accessing Page Specific Utilities**

To access Page Specific Utilities page:
1. Navigate to the Workspace home page.
2. Click the Application Builder icon.
   The Application 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: "Utilizing Cross Page Utilities" on page 7-66

Utilizing Cross Page Utilities
This section describes the tools and reports available on the Cross Page Utilities page.

- Accessing Cross Page Utilities
- Grid Edit of all Pages
- Delete Multiple Pages
- Page Attributes
- Page Groups
- Page Locks

See Also: "Accessing Application Utilities" on page 7-61

Accessing Cross Page Utilities
To access the Cross Page Utilities page:

1. On the Workspace home page, click the Application 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. See "Grid Edit of all Pages" on page 7-67.
   - Delete Multiple Pages. See "Delete Multiple Pages" on page 7-67.
Accessing Application Utilities

- **Page Attributes.** See "Page Attributes" on page 7-67.

**Grid Edit of all Pages**
Use the Grid Edit of all Pages 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**
Use Delete Multiple Pages to delete multiple pages at once. See "Deleting Multiple Pages" on page 9-27.

**Page Attributes**
The Page Attributes report details attributes, components, controls, and application logic for each page in the current application. To customize the report, use the search bar at the top of the page. See "About the Search Bar" on page 10-57.

To link to a specific page, click the page number.

**Page Groups**
Use Page Groups to better organize the pages within an application. See "Creating Page Groups" on page 7-50.

**Page Locks**
Use Page Locks to prevent conflicts during application development. By locking a page, you prevent other developers from editing it. See "Using Page Designer" on page 8-14 and "Locking and Unlocking a Page" on page 7-52.

**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 Application Builder icon.
   The Application 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.

**Using Interactive Report Utilities**
This section describes options under Interactive Report Utilities on the right side of the Utilities page.
Deleting Saved Interactive Reports
The Saved Interactive Reports page lists saved interactive reports in the currently selected application.

To manage saved interactive reports:

1. Sign in to Oracle Application Express.
   The Workspace home page appears.
2. Click the Application Builder icon.
   The Application Builder home page appears.
3. Select an application.
   The Application home page appears.
4. Click Utilities.
5. 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.
6. To view private saved reports, deselect the checkbox for the Primary Default filter located below the search bar.
7. To delete a report, select the check box adjacent to the page number and click Delete Checked.
8. To view the page containing the report, click the page number.

See Also:  "Linking to Shared Interactive Reports" on page 10-32, "Saving Interactive Reports" on page 10-60, and "Managing Saved Interactive Reports" in Oracle Application Express Administration Guide

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. Sign in to Oracle Application Express.
   The Workspace home page appears.
2. Click the Application Builder icon.
   The Application Builder home page appears.
3. Select an application.
   The Application home page appears.
4. Click Utilities.
   A report appears.
Managing Pages in Page Designer

You can view and manage application pages in Application Builder in two ways: Page Designer or Component view. This section explains how to manage database application pages in Page Designer.

- About the Page Designer Window
- Understanding Common Page Designer UI Elements
- Using Page Designer
- Editing Page Attributes in Page Designer
- Running a Page in Page Designer
- Runtime Developer Toolbar
- Copying a Database Application Page in Page Designer
- Deleting Pages in Page Designer

See Also: "Switching from Page Designer to Component View" on page 2-7, "Adding Developer Comments" on page 7-56, "Creating Database Applications" on page 7-1, "Controlling Page Layout" on page 14-1, and "Adding Navigation" on page 16-1

About the Page Designer Window

Page Designer is a full featured Integrated Development Environment (IDE) designed to maximize developer productivity for maintaining application pages.
As shown in the previous illustration, the Page Designer window is divided into three panes:

- **Left Pane** - The left pane features the following tabs: Rendering, Dynamic Actions, Processing, and Page Shared Components. See "Tree Pane" on page 8-5.
- **Central Pane** - The central pane features the following tabs: Grid Layout and associated Gallery, Messages, Page Search and Help. See "Central Pane" on page 8-6.
- **Right Pane** - The Property Editor displays in the right pane. Use the Property Editor to edit the properties of the component currently selected in the Tree pane or Grid Layout. See "Property Editor" on page 8-11.

**Tip:** The Rendering tree and Grid Layout are two different representations of the regions, items, and buttons that define a page.

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 in the center of each splitter.

**See Also:** "Understanding Common Page Designer UI Elements" on page 8-2

### Understanding Common Page Designer UI Elements

The section describes common UI elements of Page Designer.

- Page Designer Toolbar
- Tree Pane
- Central Pane
- Grid Layout
Page Designer Toolbar

The Page Designer 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. The Page Designer toolbar features the following buttons and menu options:

- **Page Finder.** Displays the Page Finder to search for pages. See "Searching for Pages" on page 8-18.
- **Page Unlocked and Page Locked.** The Page Unlocked and Page Locked button indicates whether a page is available for editing. If a page is unlocked, the button appears as an open padlock. If the page is locked, the button appears as a locked padlock and the tooltip includes the user who locked the page. A green locked padlock indicates that you have locked the page. A red locked padlock indicates another developer has locked the page. To change the lock status of a page or update the lock comment, click this button. See "Locking and Unlocking a Page" on page 7-52.
- **Undo.** Applies to actions that result in a change to the page data. Click **Undo** to revert the previous change.
- **Redo.** Applies to actions that result in a change to the page data. Click **Redo** to repeat the previously reverted change.
- **Create:**
  - **Page.** See "About Adding Pages to a Database Application" on page 7-36.
  - **Page as Copy.** See "Copying a Database Application Page in Page Designer" on page 8-25.
  - **Form Region.** See "Creating a Form in Page Designer" on page 11-6.

**Tip:** Developer Navigation tools display in the header region on many Oracle Application Express pages. To learn more, see "Developer Navigation Tools" on page 1-13.

**See Also:** "Using Page Designer" on page 8-14
- **Page Component.** Provides a summary of how to create page components in Page Designer.
- **Shared Component.** See "Managing Shared Components" on page 18-1.
- **Page Group.** See "Creating Page Groups" on page 7-50.
- **Developer Comment.** See "Adding Developer Comments" on page 7-56.
- **Team Development.** Displays the submenus Feature, To Do, and Bug. See "Tracking Features" on page 5-3, "Tracking To Dos" on page 5-10, and "Managing Bugs" on page 5-13.

**Utilities:**
- **Delete Page.** See "Deleting Pages in Page Designer" on page 8-26.
- **Advisor.** See "Running Advisor to Check Application Integrity" on page 22-1.
- **Caching.** See "Utilizing Region Caching in Page Designer" on page 14-7 and "Utilizing Region Caching in Component View" on page 14-18.
- **Attribute Dictionary.** See "Using the Attribute Dictionary" on page 17-32.
- **History.** Displays a report of changes to the current page.
- **Export.** See "Exporting an Application and Application Components" on page 24-12.
- **Cross Page Utilities.** See "Utilizing Cross Page Utilities" on page 7-66.
- **Application Utilities.** See "Using Application Builder Utilities" on page 7-61.
- **Page Groups.** See "Creating Page Groups" on page 7-50.
- **Upgrade Application.** See "Running Upgrade Application" on page 7-64.

**Component View** - Displays the page in Component View. See "Managing Pages in Component View" on page 9-1.

**Team Development:**

**Tip:** Team Development only displays if this feature is enabled for the workspace. See "Managing Team Development" in Oracle Application Express Administration Guide.

- **Features.** Links to Create Feature. See "Tracking Features" on page 5-3.
- **To Do** - Links to Create To Do. See "Tracking To Dos" on page 5-10.

**Developer Comments** - Displays the View Developer Comments window. See "Adding Developer Comments" on page 7-56.

**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" on page 18-1.

**Save.** Saves the current page.

**Save and Run Page.** Saves and then runs the current page. See "Running a Page in Page Designer" on page 8-24.
Tree Pane

The Tree pane displays in the left pane and displays regions, page items, buttons, application logic (such as computations, processes, and validations), dynamic actions, branches, and shared components as nodes on a tree. The Tree pane features four tabs:

- **Rendering** - Displays regions, page items, page buttons, page components, and application logic as nodes in a tree.
- **Dynamic Actions** - Displays dynamic actions defined on this page. By creating a dynamic action, you can define complex client-side behavior declaratively without the need for JavaScript. See "Managing Dynamic Actions" on page 15-30.
- **Processing** - Displays application logic defined on this page. See "Managing Computations, Validations, and Processes" on page 17-1.
- **Page Shared Components** - Displays shared components associated with this page. See "Managing Shared Components" on page 18-1.

The Dynamic Actions and Page Shared Components trees are grouped by event or component type and then ordered by sequence. The Rendering and Processing trees are grouped and ordered based on how Oracle Application Express processes them. This organization enables you to better understand when a component is processed and how the page is laid out. The Rendering and Processing tree have an option to group the nodes by component type. To control how tree components display click the **Group by Processing Order** or **Group by Component Type** buttons in the tree header bar.

Key features of the Tree 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.** Reorder components and controls by dragging and dropping within the tree.
- **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.

## Central Pane

The central pane in Page Designer contains four tabs:

- **Grid Layout** - Grid Layout is a visual representation of the regions, items, and buttons that define a page. See "Grid Layout" on page 8-7. You can add new regions, items and buttons to a page by selecting them from the Gallery at the bottom of the page. See "Gallery" on page 8-9.
- **Messages** - Displays to right of Grid Layout. When you create components or edit attributes in Page Designer, Messages notifies you of errors and warnings you need to address. See "Messages" on page 8-9.
- **Page Search** - Displays to the right of Messages. Use Page Search to search all page metadata including regions, items, buttons, dynamic actions, columns, and so on. See "Page Search" on page 8-10.
- **Help** - Displays to the right of Page Search. Help displays Help text for Property Editor attributes. See "Help" on page 8-11.
Grid Layout displays in the central pane and is a visual representation of the regions, items, and buttons that define a page. You can adjust the size of Grid Layout using the window splitters. Alternatively, you can click the **Expand** button to make Grid Layout larger and then **Restore** to return it to the previous size.

The Grid 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 Grid Layout pane, select the region and select **Display from Here** from the Grid Layout menu. The selected region fills Grid Layout. To restore the view, select **Display from Page**.
You can move, copy, and reorder regions, items, and buttons by dragging and dropping. You can add new regions, items, and buttons by dragging them from the Gallery and dropping them to the desired position on the page. 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.

Key features of the Grid Layout 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.** Create new or move, copy, and reorder regions, items, and buttons by dragging and dropping.
- **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:** Hidden items do not display in Grid Layout, but do display in the Rendering tree.
Gallery

The Gallery displays at the bottom of the Grid Layout pane and features three tabs: Regions, Items, and Buttons. By default, only supported controls and components appear. When you pass the cursor over a control or component, a tooltip displays that describes it. You can add new controls and components to a page by selecting them in the Gallery and dragging and dropping them into Grid Layout.

See Also: "Adding a Region, Item, or Button by Dragging and Dropping" on page 8-15

Messages

When you create components or edit attributes in Page Designer, Messages notify you of 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. 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 tree and Grid Layout.

**See Also:** "Adding a Region, Item, or Button by Dragging and Dropping" on page 8-15

---

**Page Search**

Page Search displays to the right of Messages in the central pane. Use Page Search to search all page metadata including regions, items, buttons, dynamic actions, columns, and so on. To search a page, enter a search terms in the field provided. To match the case, select Match Case. To search for a regular expression, select Regular Expression.
Help

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.

See Also: "Viewing Help in Page Designer" on page 1-22

Property Editor

The Property Editor displays in the right pane and displays all attributes for the current component. As you select different components in either Tree View or Grid Layout, the Property Editor automatically updates to reflect the current selection. Attributes are organized into functional groups that describe their purpose. Required attributes display a red triangle in the upper left corner above the attribute label.

You can control the amount of information that displays using the following buttons at the top of the pane:
- **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** - Shows only functional group headings.
- **Expand All** - Shows all attributes under each group heading.
- **Go to Group** - Selects to the first attribute in the group you select.

**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.
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.

See Also:  "Accessing the Property Editor" on page 8-15
Key features of the Code Editor include:

- Undo
- Redo
- Find
- Replace
- Query Builder
- Autocomplete
- 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.

**Using Page Designer**

This section explains how to view and edit database application pages using the Page Designer.

- Viewing a Page in Page Designer
- Accessing the Property Editor
- Adding a Region, Item, or Button by Dragging and Dropping
- Adding a Region, Item, or Button Using Gallery Context Menus
Viewing a Page in Page Designer

To view a page in Page Designer:

1. Navigate to the Workspace home page.
2. Click the Application 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 Component View to Page Designer" on page 2-8 and "Running a Page in Page Designer" on page 8-24

Accessing the Property Editor

The Property Editor displays in the right pane and displays all attributes for the currently selected components.

To access the Property Editor:

2. In either the Tree pane or Grid Layout, select components or controls to edit.

**Tip:** Passing the cursor over a control or component, displays a tooltip that describes it.

The Property Editor displays attributes for the selected component or control.

**See Also:** "Property Editor" on page 8-11

Adding a Region, Item, or Button by Dragging and Dropping

Each 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 from the Gallery:

2. If needed, select the Grid Layout tab in the central pane.
3. From the Gallery locate the region, item or button you want to create.
4. Click and hold the mouse on the component to be created and drag it to the desired location in Grid Layout.

**See Also:** "Switching from Component View to Page Designer" on page 2-8 and "Running a Page in Page Designer" on page 8-24
Tip: Components can only be dropped into appropriate drop positions, as determined by the component type. If you wish to place the new component next to an existing component then drag the new component to the right, and level, with the existing component, and wait a short period until a new drop zone is displayed.

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. See "Messages" on page 8-9. The following example shows a new Interactive Report region with an error message indicating that a SQL statement is required.

5. Edit the appropriate attributes in the Property Editor. Required attributes display a red triangle in the upper left corner next above attribute label.

6. Click Save or Save and Run Page.

See Also: "Running a Page in Page Designer" on page 8-24 and "Adding a Region, Item, or Button Using Gallery Context Menus" on page 8-16

Adding a Region, Item, or Button Using Gallery Context Menus

In addition to dragging and dropping components from the Gallery, you can also right-click a component to view a context menu. Each context menu features an Add To option which displays the actual locations where you can add the component.
Adding a Region, Item, or Button from the Rendering Tree

To create new components within the Rendering tree:

2. In the left pane, select the Rendering tab.
3. Navigate to the appropriate region, item, or region button.
4. Right-click on an existing component or container to display the context menu for that component type.
5. Select Create Region, Create Page Item, or Create Button. The options available depend upon the node you select.
6. Edit the appropriate attributes in the Property Editor. Required attributes display a red triangle in the upper left corner next above attribute label.
7. Click Save or Save and Run Page.

See Also: "Adding a Region, Item, or Button by Dragging and Dropping" on page 8-15

Creating Pre-Rendering Branches, Computations, and Processes

To create branches, computations, and processes that execute before rendering the page:

2. In the left pane, select the Rendering tab.
3. Expand the Pre-Rending node at the top of the Rendering tree.
4. Right-click Before Header, After Header, Before Regions to display the context menu for that processing point.
5. Select Create Branch, Create Computation, or Create Process.
6. Edit the appropriate attributes in the Property Editor. Required attributes display a red triangle in the upper left corner next above attribute label.
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" on page 17-1

### Creating Page Submission Branches, Validations, Computations, and Processes

To create branches, validations, computations, and processes that execute when the page is submitted:

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. Required attributes display a red triangle in the upper left corner next above attribute label.
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" on page 17-1

### Searching for Pages
The Page Finder is the first control in the Page Designer toolbar and displays the current page number. Use the Page Finder to search for pages within the current application.

To search for pages within an application:
2. On the Page Designer toolbar, click the Page Finder. The Page Finder appears.
3. Click the tabs at the top of the window to change the view:
   - **Current User Interface** - Displays pages in the current user interface.
   - **Current Group** - If the current page is associated with a group, this tab displays the pages associated with the current group.
   - **All Pages** - Displays all pages in the current application.
   - **Recently Edited** - Displays recently edited pages.
4. Enter search terms in the Search field and press ENTER. 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 in Page Designer" on page 8-24 and "Creating Page Groups" on page 7-50

**Editing Page Attributes in Page Designer**
This section describes how to view and edit page attributes in the Property Editor in Page Designer. Common page attributes you may want to edit include the page name or title, header and footer text, and the selected authorization scheme.

- **Viewing Page Attributes in the Property Editor**
- **Property Editor - Page**
- **Incorporating JavaScript Using Page Designer**
- **Configuring Rejoin Sessions in Page Designer**
- **Configuring Dialogs Attributes in Page Designer**

**Viewing Page Attributes in the Property Editor**
To view page attributes in the Property Editor:
2. In either the Rendering Tree or Grid Layout, select the page name. The page name is selected by default when editing a page. The Property Editor displays the Page attributes.
3. Edit the appropriate attributes.
To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane. Once you activate the Help pane, the content that displays changes every time you select another attribute.

**See Also:** "Viewing Help in Page Designer" on page 1-22

**Property Editor - Page**

By default, Property Editor displays all attributes.

You can control how attributes display using the following controls at the top of the pane:

- **Show Common** - Displays only those attributes for which the default value has changed or are commonly edited.
- **Show All** - Displays all attributes.
- **Collapse All** - Shows only functional group headings.
- **Expand All** - Shows all attributes under each group heading.

Required attributes display a red triangle in the upper left corner above the attribute label.

To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane. Once you activate the Help pane, the content that displays changes every time you select another attribute.

**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 attributes in the Property Editor. "Viewing Page Attributes in the Property Editor" on page 8-19.
2. Click the Collapse All icon and then locate and expand JavaScript.
3. 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 center 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.

     **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.

     ```javascript
     function test(){
         alert('This is a test. ');
     }
     ```

4. Click Save.

**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" on page 20-5.
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 attributes in the Property Editor. "Viewing Page Attributes in the Property Editor" on page 8-19.
2. Click the Collapse All icon and then locate and expand Security.
3. From Rejoin Sessions, select an option:
   - **Application Default** - Inherits the setting defined at application-level. See "Browser Security" on page 7-24.
   - **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.
4. Click Save.

**See Also:** "Browser Security" on page 7-24, "About Rejoin Sessions" on page 20-5, and "Configuring Rejoin Sessions" in Oracle Application Express Administration Guide

### Configuring Dialogs Attributes in Page Designer

**Tip:** Creating a dialog page is a multiple step process. To learn more, see "Creating Dialog Pages" on page 7-40.

To configure Dialog attributes in Page Designer:

1. View the page attributes in the Property Editor. "Viewing Page Attributes in the Property Editor" on page 8-19.
2. Click the Collapse All icon.
3. Expand Appearance and verify that the Page Mode is set to either **Modal Dialog** or **Non-Modal Dialog**. See field-level Help for more information.
4. Expand **Dialog**, and configure the attributes described in **Table 8–1**.

**Table 8–1 Page Attributes: Dialog**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>Enter the width of the dialog, in pixels or a percentage. This setting overwrites the default dialog width specified at page template-level. 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%. The substitution string #DIALOG_WIDTH# in the dialog initialization code of the page template is replaced with the value entered in this item.</td>
</tr>
<tr>
<td>Height</td>
<td>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 value entered in this item.</td>
</tr>
<tr>
<td>Maximum Width</td>
<td>Enter the maximum width of the dialog, in pixels. This setting will overwrite the default dialog maximum width specified at page template-level. 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.</td>
</tr>
<tr>
<td>Attributes</td>
<td>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. 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, replaces any values entered in this item. For more information, see Help:</td>
</tr>
<tr>
<td>CSS Classes</td>
<td>Enter additional CSS classes you wish to be applied to the dialog, as a space delimited list. For example: ui-dialog--wizard Any occurrence of the substitution string #DIALOG_CSS_CLASSES# in the page template is replaced with the values specified in this field. This value overwrites any value set at page template-level.</td>
</tr>
<tr>
<td>Chained Dialog</td>
<td>Select <strong>Yes</strong> 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 <strong>No</strong> 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.</td>
</tr>
</tbody>
</table>

5. Click **Save**.
Running a Page in 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:


   You preview a page by running it. Running an application page displays the rendered version as end users would see it.

2. Click the **Save and Run Page** button in the upper right corner. The Save and Run Page button resembles a blue right-facing arrow.

A rendered version of the application page appears.

**See Also:** "Creating Dialog Pages" on page 7-40

**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.

**See Also:** "Running an Application or Page" on page 2-9 and "Runtime Developer Toolbar" on page 8-24

Runtime Developer Toolbar

When you run a desktop application from Application Builder, 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. 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 consists of the following controls:

- **Home** links to the Workspace home page. See "Understanding the Workspace Home Page" on page 1-10.
- **Application** links to the Application home page. See "Using the Application Builder Home Page" on page 4-2.
■ **Edit Page** displays the currently running page ID. The View that appears depends upon your current preference. See "Understanding the Difference Between Page Designer and Component View" on page 2-4.

■ **Session** links you to session state information for the current page. See "Viewing Session State" on page 2-15.

■ **View Debug** displays the Debug reports. See "Debugging an Application" on page 22-1.

■ **Debug** toggles the page between Debug and No Debug mode. See "Utilizing Debug Mode" on page 22-3.

■ **Show Grid** toggles between Show Grid and Hide Grid. This option only applies if a grid layout is used and that layout supports showing a grid and the page has more than one column.

■ **Quick Edit** enables developers to enter Quick Edit mode and then select the desired component to instantly access that component in Page Designer. Press ESCAPE or click outside a component to exit quick edit mode.

■ **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" on page 13-3.

■ **Developer Toolbar Options** displays on the far right and resembles a mechanical gear. You can customize how the Developer Toolbar displays by using the following controls:
  - Auto Hide
  - Show Icons Only
  - Display Position (Top, Left, Bottom, Right)

---

**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. Navigate to the page you want to copy. See "Viewing a Page in Page Designer" on page 8-15.

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:
   - Page in this application
   - Page in another application
4. Follow the on-screen instructions.

Deleting Pages in Page Designer
Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries.

- Deleting a Page in Page Designer
- Deleting Multiple Pages
- Deleting a Range of Pages

See Also: "Locking and Unlocking a Page" on page 7-52

Deleting a Page in Page Designer
To delete a page in Page Designer:
2. Verify the page number.
4. Click Permanently Delete Page.

See Also: "Page Designer Toolbar" on page 8-3

Deleting Multiple Pages
To delete multiple pages:
1. On the Workspace home page, click the **Application Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities, click **Cross Page Utilities**.
5. Click **Delete Multiple Pages**.
6. Select the pages to be deleted and click **Delete Pages**.

**Deleting a Range of Pages**

To delete a range of pages:

1. On the Workspace home page, click the **Application 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**.
You can view and manage application pages in Application Builder in two ways: Page Designer or Component view. This section explains how to manage database application pages in Component View:

- About the Component View Window
- Understanding Component View Common UI Elements
- Using Component View
- Editing Page Attributes in Component View
- Running an Application or Page in Component View
- Runtime Developer Toolbar
- Copying a Database Application Page in Component View
- Deleting Pages in Component View

See Also: "Switching from Page Designer to Component View" on page 2-7, "Adding Developer Comments" on page 7-56, "Creating Database Applications" on page 7-1, "Controlling Page Layout" on page 14-1, and "Adding Navigation" on page 16-1
About the Component View Window

Component View is the traditional user interface for editing a page in Application 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.

The 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 Component View" on page 9-5 and “Editing a Page in Component View” on page 9-5

Understanding Component View Common UI Elements

The section describes common Component View UI elements.
Understanding Component View Common UI Elements

- Component View Toolbar
- Navigation Bar


**See Also:** "Viewing a Page in Component View" on page 9-5 and "Editing a Page in Component View" on page 9-5

### Component View Toolbar

![Component View Toolbar](image)

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. The Page Designer 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. See "Adding Developer Comments" on page 7-56, "Managing Bugs" on page 5-13, and "Tracking To Dos" on page 5-10.

- **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. See "Using the Find Icon" on page 7-45.

- **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" on page 18-1.

- **Edit Page.** Displays the current page.

- **Run Page.** Runs the current page. See "Running an Application or Page in Component View" on page 9-24.

### 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 Finder. Use the Page Finder to navigate between pages. To navigate to page, enter a page number in the Page Finder 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. See "About Adding Pages to a Database Application" on page 7-36.
- Region. See "Creating a Region in Component View" on page 14-14.
- Page Component. See "Creating a Page Control Component View" on page 9-10.
- Shared Components. See "Managing Shared Components" on page 18-1.
- Developer Comment. See "Adding Developer Comments" on page 7-56.
- Team Development. Displays the submenus Feature, To Do, and Bug. See "Tracking Features" on page 5-3, "Tracking To Dos" on page 5-10, and "Managing Bugs" on page 5-13.

Utilities:
- Advisor. See "Running Advisor to Check Application Integrity" on page 22-1.
- Resequence Components.
- Clean up. Re-sequences the items on the page in increments of 10.
- Lock. See "Locking and Unlocking a Page" on page 7-52.
- 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.
- Caching. See "Utilizing Region Caching in Page Designer" on page 14-7 and "Utilizing Region Caching in Component View" on page 14-18.
Using Component View

This section explains how to view and edit database application pages using Component View.

- **History.** Displays a report of changes to the current page.
- **Export.** See "Exporting an Application and Application Components" on page 24-12.
- **Cross Page Utilities.** See "Utilizing Cross Page Utilities" on page 7-66.
- **Application Utilities.** See "Using Application Builder Utilities" on page 7-61.
- **Page Groups.** See "Viewing Page Groups" on page 7-50.
- **Upgrade Application.** See "Running Upgrade Application" on page 7-64.

■ **Page Designer** - Click **Page Designer** to view the page in Page Designer. See "Managing Pages in Page Designer" on page 8-1.

Using Component View

This section explains how to view and edit database application pages using Component View.

- **Viewing a Page in Component View**
- **Editing a Page in Component View**
- **Understanding Page Rendering**
- **Understanding Page Processing**
- **Understanding Shared Components**
- **Creating a Page Control Component View**

**Viewing a Page in Component View**

To view a page in Component View:

1. Navigate to the Workspace home page.
2. Click the **Application Builder** icon.
3. Select an application.
4. Select a page.
   
   **Tip:** If you previously viewed pages in Component View, Component View displays by default.

5. On the Page Designer toolbar, click the **Component View** icon. Component View appears.
   
   You preview a page by running it. Running an application page displays the rendered version as end users would see it.

**Editing a Page in Component View**

This section explains how to edit a page in Component View.

- **About Using Component View**
- **Edit All Icon**
- **Copy or Create Icons**
About Using Component View

In Component View each page is divided into three sections: Page Rendering, Page Processing, and Shared Components. Each of these sections is broken into subsections with headings that identify a type of control, component, or application logic.

To edit a component, drill down on the component name or click the icon described in the following sections.

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.

Copy or Create Icons

You can copy or create controls or components by clicking the Copy and Create icons. The Copy icon resembles two small overlapping pages. Click the Copy icon to make a copy of an existing control or component.
Using Component View

Managing Pages in Component View

The Create icon resembles a plus (+) sign. Click the Create icon to create a control or component.

Understanding Page Rendering

This section describes the subsections under Page Rendering in Component View.

- What Is Page Rendering?
- Page Rendering

What Is Page Rendering?

Page rendering is the process of generating a page from the database. Developers use the Page Rendering section to modify controls that impact the rendering of a page, including page attributes, regions, buttons, items, and page processes.

Page Rendering

Table 9–1 describes the subsections under Page Rendering in Component View.
Table 9–1  Page Rendering Subsections

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Description</th>
</tr>
</thead>
</table>
| Page       | 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.  

See Also: “Editing Page Attributes in Component View” on page 9-11 |
| Regions    | 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.  

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.  

See Also: “About Regions” on page 14-4, Oracle Database Development Guide for information about developing web applications with PL/SQL, and Oracle Database PL/SQL Packages and Types Reference for information about htp packages |
| Buttons    | 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.  

See Also: “Creating Buttons” on page 15-42 |
| Items      | 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.  

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.  

See Also: “Understanding Page-Level Items” on page 15-1 and “Managing Application-Level Items” on page 18-15 |
| 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” on page 17-17 and “Understanding Application Processes” on page 17-25 |
| 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” on page 15-30 |

Understanding Page Processing

This section describes the subsections under Page Processing in Component View.

- What Is Page Processing?
- Page Processing
What Is 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.

Page Processing
Table 9–2 describes the subsections under Page Processing in Component View.

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computations</td>
<td>Computations are units of logic used to assign session state to items and are executed at the time the page is processed. See Also: &quot;Understanding Page Computations&quot; on page 17-1</td>
</tr>
<tr>
<td>Processes</td>
<td>Processes are logic controls used to execute Data Manipulation Language (DML) or PL/SQL. Processes are executed after the page is submitted. See Also: &quot;Understanding Page Processes&quot; on page 17-17</td>
</tr>
<tr>
<td>Branches</td>
<td>Branches enable you to create logic controls that determine how the user navigates through the application. See Also: &quot;Controlling Navigation Using Branches&quot; on page 16-20 and &quot;About the When Button Pressed Attribute&quot; on page 2-39</td>
</tr>
</tbody>
</table>

Understanding Shared Components
This section describes the subsections that may display under Shared Components in Component View.

- What Are Shared Components?
- Shared Components

See Also: "Managing Shared Components" on page 18-1

What Are Shared Components?
The Shared Components section contains common elements that can display or be applied on any page within an application.

See Also: "Managing Shared Components" on page 18-1

Shared Components
Table 9–3 describes the subsections under Page Processing in Component View.
Creating a Page Control Component View

Page controls include items, buttons, branches, computations, processes, or validations.

To add a page control in Component View:

**Table 9–3  Shared Components Subsections**

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tabs</td>
<td>Tabs are an effective way to navigate between pages of an application. Application Builder includes two types of tabs: standard tabs and parent tabs. 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. <strong>See Also: “Creating Tabs” on page 16-1</strong></td>
</tr>
<tr>
<td>List of Values</td>
<td>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. You define LOVs at the application level by running the LOV Wizard and adding them to the List of Values repository. <strong>See Also: “Creating Lists of Values at the Application Level” on page 18-18</strong></td>
</tr>
<tr>
<td>Breadcrumbs</td>
<td>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. <strong>See Also: “Creating Breadcrumbs” on page 16-7</strong></td>
</tr>
<tr>
<td>Lists</td>
<td>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. <strong>See Also: “Creating Lists” on page 20</strong></td>
</tr>
<tr>
<td>Theme</td>
<td>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. <strong>See Also: “Using Themes” on page 13-1</strong></td>
</tr>
<tr>
<td>Templates</td>
<td>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. <strong>See Also: “Creating Custom Themes” on page 13-23</strong></td>
</tr>
<tr>
<td>Security</td>
<td>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. <strong>See Also: “Providing Security Through Authorization” on page 20-49</strong></td>
</tr>
<tr>
<td>Navigation Bar</td>
<td>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). <strong>See Also: “Creating a Navigation Bar Entry” on page 16-15</strong></td>
</tr>
</tbody>
</table>

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. See "Understanding Page-Level Items" on page 15-1.
   - Computation. See "Understanding Page Computations" on page 17-1.
   - Validation. See "Understanding Validations" on page 17-6.

4. Follow the on-screen instructions.

**Editing Page Attributes in Component View**

This section describes how to edit page attributes in 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.

- Viewing Page Attributes in Component View
- Page Attributes in Component View
- Incorporating JavaScript Using Component View
- Configuring Rejoin Sessions in Component View

**Viewing Page Attributes in Component View**

To edit page attributes:


2. Under Page, click the Edit icon.
   
The Page Attributes page appears. Required values are marked with a red asterisk (*).

3. Edit the appropriate attributes.
   
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.

**Page Attributes in Component View**

This section describes the regions on the Page Attributes page.

- Name
Display Attributes
JavaScript
CSS
Dialog
HTML Header
Header and Footer
Read Only
Security
Duplicate Submission
Server Cache
Configuration
Error Handling
Help
Comments

Name
Table 9–4 describes general attributes for the current page such as a page name, an optional alphanumeric alias, and associated page groups.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>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.</td>
</tr>
<tr>
<td>Page Alias</td>
<td>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 home. You could then access this page from other pages using the following f?p syntax: f?p=100:home</td>
</tr>
<tr>
<td>Group</td>
<td>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.</td>
</tr>
</tbody>
</table>

See Also: "Creating Page Groups" on page 7-50

Display Attributes
Table 9–5 describes general display attributes for the current page such as the selected page template, standard tab set, title, and cursor focus.
Table 9–6 describes attributes that enable you to include JavaScript on the current page.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>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 #TITLE# substitution string used in the page template. This title is inserted between the HTML tags &lt;TITLE&gt; and &lt;/TITLE&gt;.</td>
</tr>
<tr>
<td>Page Mode</td>
<td>Select a page mode. Options include:</td>
</tr>
<tr>
<td></td>
<td>■ Normal - The page displays as a normal Application Express application page.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td>Page Template Options</td>
<td>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.</td>
</tr>
<tr>
<td>Page CSS Classes</td>
<td>Enter additional CSS classes as space a delimited list which are substituted if the page template contains the substitution string #PAGE_CSS_CLASSES#.</td>
</tr>
<tr>
<td>Cursor Focus</td>
<td>Specify the cursor focus. Available options include:</td>
</tr>
<tr>
<td></td>
<td>■ Select First item on page to have the cursor focus placed in the first field on the page.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td>Media Type</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

**JavaScript**

Table 9–6 describes attributes that enable you to include JavaScript on the current page.
Table 9–6  JavaScript Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File URLs</td>
<td>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.</td>
</tr>
</tbody>
</table>

To view examples, see field-level Help.

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.                                                                                     |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Code you enter here replaces the #PAGE_JAVASCRIPT# substitution string in the page template.</td>
<td></td>
</tr>
<tr>
<td>Note: Do not to include opening or closing script tags. Just include the JavaScript code.</td>
<td></td>
</tr>
<tr>
<td>To reference a shortcut, use:</td>
<td>&quot;SHORTCUTNAME&quot;</td>
</tr>
</tbody>
</table>

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.                                                                                       |                                                                                                                                                                                                                                                                                                                                                     |
| Note: Do not to add any opening or closing script tags, just the JavaScript code.                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                     |

CSS

Table 9–7 describes attributes that enable you to include Cascading Style Sheet (CSS) files on the current page.

Table 9–7  CSS Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File URLs</td>
<td>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.</td>
</tr>
</tbody>
</table>

File URLs you enter here replace the #PAGE_CSS# substitution string in the page template.                                                                                                                                                                                                                                                          |

Note: You do not need to include opening or closing link tags. Just include the file URL.                                                                                                                                                                                                                                                                                                                            |

To view examples, see field-level Help.
Dialog

Table 9–8 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" on page 13-45.

Table 9–8 Page Attributes: Dialog

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>Enter the width of the dialog, in pixels or a percentage. This setting overwrites the default dialog width specified at page template-level. 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%. 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.</td>
</tr>
<tr>
<td>Height</td>
<td>Enter the height of the dialog in pixels, for example, 500. The substitution string <code>#DIALOG_HEIGHT#</code> in the dialog initialization code of the page template is replaced with value entered in this item.</td>
</tr>
<tr>
<td>Maximum Width</td>
<td>Enter the maximum width of the dialog, in pixels. This setting will overwrite the default dialog maximum width specified at page template-level. 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.</td>
</tr>
<tr>
<td>Attributes</td>
<td>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. The substitution string <code>#DIALOG_ATTRIBUTES#</code> in the dialog initialization code of the page template level, will be replaced by any value(s) entered in this item. To learn more, see field-level Help.</td>
</tr>
</tbody>
</table>
Editing Page Attributes in Component View

Table 9–9 describes attributes for defining the page header and page HTML body text.

**Table 9–9 HTML Header Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML Header</td>
<td>Enter text to replace the #HEAD# substitution string in the page template header. The values entered here are inserted after the HTML &lt;HEAD&gt; tag. Use this page attribute to add page specific meta tags.</td>
</tr>
<tr>
<td>Note: JavaScript code or Cascading Style Sheets should be added to the dedicated JavaScript and CSS attributes.</td>
<td></td>
</tr>
<tr>
<td>Page HTML Body Attribute</td>
<td>Enter text to be substituted in for your page template’s #ONLOAD# substitution string. To use this feature, your page template must include the #ONLOAD# substitution string.</td>
</tr>
</tbody>
</table>

Table 9–10 describes attributes for defining the header and footer text.

**Table 9–10 Header Text and Footer Text**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Text</td>
<td>Enter any text or HTML to appear immediately following the page header and before the body content.</td>
</tr>
<tr>
<td>Footer Text</td>
<td>Enter text or HTML to appear before the page template footer.</td>
</tr>
</tbody>
</table>

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**.
Security

Table 9–11 describes attributes for defining authorization scheme and authentication requirements for the current page.

Table 9–11  Page Attributes: Security

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization Scheme</td>
<td>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. 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. <strong>See Also:</strong> &quot;Providing Security Through Authorization&quot; on page 20-49</td>
</tr>
<tr>
<td>Authentication</td>
<td>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. <strong>See Also:</strong> &quot;Establishing User Identity Through Authentication&quot; on page 20-37</td>
</tr>
<tr>
<td>Deep Linking</td>
<td>Use this attribute to enable or prevent deep linking to an application page. Options include:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Enabled</strong> - The URL to a specific page ultimately redirects there, possibly after the user has logged in.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Disabled</strong> - If the URL does not contain a valid session ID, Application Express starts a new session and redirects to the application’s home page.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Application Default</strong> - The application-level setting is used. <strong>See Also:</strong> &quot;Authentication&quot; on page 7-20</td>
</tr>
</tbody>
</table>

Page Access Protection Note: This attribute only displays if Session State Protection has been enabled for the application. See “Understanding Session State Protection” on page 20-21.

Select a Page Access Protection level for the current page:

- **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**

Setting this attribute to **On** has no effect on the page. Setting this value to **Off** generates the following HTML in the FORM tag:

```
autocomplete="off"
```

**Browser Cache**

Note: This feature requires a modern browser that supports the HTTP header response variable `cache-control`.

Options include:

- **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 `cache-control: no-store` which instructs the browser not to 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).

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 requests 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.

### Table 9–11 (Cont.) Page Attributes: Security

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form Auto Complete</td>
<td>Setting this attribute to <strong>On</strong> has no effect on the page. Setting this value to <strong>Off</strong> generates the following HTML in the FORM tag: autocomplete=&quot;off&quot;</td>
</tr>
<tr>
<td>Browser Cache</td>
<td>Note: This feature requires a modern browser that supports the HTTP header response variable <code>cache-control</code>. Options include:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Application Default</strong> - The value at runtime is taken from the application’s Browser Cache attribute.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Enabled</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Disabled</strong> - 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 not to 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).</td>
</tr>
</tbody>
</table>

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 requests 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.
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.

---

**Table 9–11 (Cont.) Page Attributes: Security**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| Rejoin Sessions    | 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. **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" on page 20-5. From Rejoin Sessions, select an option:  
- Application Default - Inherits the setting defined at application-level. See "Browser Security" on page 7-24.  
- 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:  
  - 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.  

Server Cache

Table 9–12 describes attributes to enable caching for the current page.

Table 9–12  Page Attributes: Server Cache

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Page</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>■ Disabled - Content is not cached and computed for each request.</td>
</tr>
<tr>
<td></td>
<td>■ Enabled - Content is cached and utilized by all users.</td>
</tr>
<tr>
<td></td>
<td>■ Cached by User - Content is cached specifically for each user.</td>
</tr>
<tr>
<td></td>
<td>■ Cache by Session - Content is cached specifically for each session.</td>
</tr>
<tr>
<td>Cache Timeout</td>
<td>Displays if you select Enabled or Cached by User. Specify the amount of time for which a cached page remains valid.</td>
</tr>
<tr>
<td>Cache Page Condition</td>
<td>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.</td>
</tr>
<tr>
<td>Expression 1</td>
<td>Displays conditionally based on the selected Cache Page Condition. Enter values in this attribute based on the specific condition type selected.</td>
</tr>
<tr>
<td>Expression 2</td>
<td>Displays conditionally based on the selected Cache Page Condition. Enter values in this attribute based on the specific condition type selected.</td>
</tr>
</tbody>
</table>

See Also:  "Managing Cached Regions and Pages" in Oracle Application Express Administration Guide and "Utilizing Region Caching in Component View" on page 14-18

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" on page 24-31

Error Handling

Table 9–13 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 9–13  Page Attributes: Error Handling

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-line Error Notification Text</td>
<td>Enter the error text that displays in the #NOTIFICATION_MESSAGE# template substitution string in the event an error occurs on the page.</td>
</tr>
</tbody>
</table>
| Error Handling Function       | 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:  

```plsql
log_apex_error
```

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:

```plsql
#OWNER#.log_apex_error
```

You must implement error handling functions using the syntax described in the apex_error package.

```plsql
function <name of function> ( 
p_error in apex_error.t_error )  
return apex_error.t_error_result
```

See Also: apex_error in Oracle Application Express API Reference

**Note:** Error handling specified at the page-level overwrites any error handling function specified at the application-level. See "Editing the Application Attributes" on page 7-14.

---

**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:

2. Under Page, click the Edit All icon.

---

**See Also:**  "Page Templates" on page 13-42

**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" on page 12-39 and "Using Shortcuts" on page 18-34

**Comments**

Use this attribute to record comments about the current page. These comments never display when the application is running.
3. Under JavaScript, enter code into the following attributes:

**Tip:** To learn more about an attribute, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

- **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.

```javascript
function test()
    alert('This is a test.');
```

4. Click **Apply Changes**.

### 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" on page 20-5.

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:


2. Under Page, click the **Edit All** icon.


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**.

### Configuring Dialog Attributes in Component View

**Tip:** Creating a dialog page is a multiple step process. To learn more, see "Creating Dialog Pages" on page 7-40.

To configure Dialog attributes 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 described in **Table 9–14**.

### Table 9–14  Page Attributes: Dialog

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Width</strong></td>
<td>Enter the width of the dialog, in pixels or a percentage. This setting overwrites the default dialog width specified at page template-level.</td>
</tr>
<tr>
<td></td>
<td>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%.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Height</strong></td>
<td>Enter the height of the dialog in pixels, for example, 500.</td>
</tr>
<tr>
<td></td>
<td>The substitution string <code>#DIALOG_HEIGHT#</code> in the dialog initialization code of the page template is replaced with value entered in this item.</td>
</tr>
<tr>
<td><strong>Maximum Width</strong></td>
<td>Enter the maximum width of the dialog, in pixels. This setting will overwrite the default dialog maximum width specified at page template-level.</td>
</tr>
<tr>
<td></td>
<td>For a page using the jQuery Mobile Smartphone user interface, the maximum width is 500.</td>
</tr>
</tbody>
</table>
|                  | The substitution string `#DIALOG_MAX_WIDTH#` in the dialog initialization code of the page template is replaced with the value entered in this item.
Running an Application or Page in 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:


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: "Running an Application or Page" on page 2-9

---

**Table 9–14 (Cont.) Page Attributes: Dialog**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>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. The substitution string #DIALOG_ATTRIBUTES# in the dialog initialization code defined at page template-level, replaces any values entered in this item. To learn more about an attribute, see field-level Help.</td>
</tr>
<tr>
<td>CSS Classes</td>
<td>Enter additional CSS classes you wish to be applied to the dialog, as a space delimited list. For example: ui-dialog--wizard Any occurrence of the substitution string #DIALOG_CSS_CLASSES# in the page template is replaced with the values specified in this field. This value overwrites any value set at page template-level.</td>
</tr>
<tr>
<td>Chained Dialog</td>
<td>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.</td>
</tr>
</tbody>
</table>

6. Click Apply Changes.

See Also: "Creating Dialog Pages" on page 7-40
Runtime Developer Toolbar

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 by clicking the Run button.

When you run a desktop application from within the development environment, the Runtime Developer toolbar displays at the bottom of the page by default. The Runtime Developer toolbar offers a quick way to access the current application, edit the current page, view session state, or toggle in and out of Debugging mode. The Runtime Developer toolbar only displays in Availability, Status attribute on the Application Definition page is set to Available with Developer Toolbar.

The Runtime Developer toolbar consists of the following controls:

- **Home** links to the Workspace home page. See "Understanding the Workspace Home Page" on page 1-10.
- **Application** links to the Application home page. See "Using the Application Builder Home Page" on page 4-2.
- **Edit Page** displays the definition for the currently running page. The View that appears depends upon your current preference. See "Understanding the Difference Between Page Designer and Component View" on page 2-4.
- **Session** links you to session state information for the current page. See "Viewing Session State" on page 2-15.
- **View Debug** displays the Debug reports. See "Debugging an Application" on page 22-1.
- **Debug** toggles the page between Debug and No Debug mode. See "Utilizing Debug Mode" on page 22-3.
- **Show Grid** only applies if a grid layout is used and that layout supports showing a grid and the page has more than one column.
- **Quick Edit** enables developers to enter Quick Edit mode and then select the desired component to instantly access that component in Page Designer. Press ESCAPE or click outside a component to exit quick edit mode.
- **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" on page 13-3.
- **Developer Toolbar Options** displays to the right of Theme Roller and enables developers to customize how the Developer Toolbar displays by using the following controls:
  - **Auto Hide**
  - **Show Icons Only**
  - **Display Position** (Top, Left, Bottom, Right)

**Note:** The Runtime Developer toolbar does not display on locked packaged applications pages or on jQuery Mobile Smartphone pages.
Copying a Database Application Page in 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:
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.

Deleting Pages in 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
- Deleting a Page While Editing Page Attributes
- Deleting Multiple Pages
- Deleting a Range of Pages

See Also: "Locking and Unlocking a Page" on page 7-52

Deleting a Page in Component View

To delete a 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.

See Also: "Using Component View" on page 9-5

Deleting a Page While Editing Page Attributes

To delete a page while editing page attributes:
2. Verify the page number and page name.
3. Click Delete.
4. Follow the on-screen instructions.

See Also: "Using Component View" on page 9-5
Deleting Multiple Pages

To delete multiple pages:

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.

Deleting a Range of Pages

To delete a range of pages:

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.
This section describes how to generate reports in Oracle Application Express. In Oracle Application Express, a report is the formatted result of a SQL query. You can generate a report in a database application by selecting and running a built-in query, or by defining a report region based on a SQL query.

- Understanding the Difference Between Interactive and Classic Reports
- Creating Reports Using a Wizard
- Creating a Report by Dragging and Dropping from the Gallery
- Editing Interactive Reports in Page Designer
- Editing Interactive Reports in Component View
- Customizing Interactive Reports as an End User
- Editing Classic Reports in Page Designer
- Editing Classic Reports in Component View
- Printing Report Regions

See Also: "Developing Forms" on page 11-1 and "Managing Database Application Components" on page 12-1

Understanding the Difference Between Interactive and Classic Reports

When creating a database application, you can include two types of reports: an interactive report or a classic report. The main difference between these two report types is the extent to which end users can customize the appearance of the data through searching, filtering, sorting, column selection, highlighting, and other data manipulations.

- Interactive Reports
- Classic Reports

Interactive Reports

The following is an example of an interactive report.
To create an interactive report, developers either select a table or provide a SQL statement. Developers 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.

When defining an interactive report you can:

- Include multiple interactive reports per page
- Display 100 columns using report columns.
- Include 999 rows per column heading filter (if no custom LOV is specified in the column attributes)

See Also: "Customizing Interactive Reports as an End User" on page 10-57, "Editing Classic Reports in Page Designer" on page 10-64, and "Editing Interactive Reports in Component View" on page 10-33

### Classic Reports

In contrast, with the exception of sorting and simple filtering, end users cannot customize a classic report. The following is an example of a classic report built on top of the same data.
Creating Reports Using a Wizard

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.

When using a classic report, you can view 100 columns using report columns.

**Tip:** Developers can create highly customized reports using report template (in particular, named column templates). See “Report Templates” on page 13-51.

**See Also:** "Editing Classic Reports in Page Designer” on page 10-64, "Editing Classic Reports in Component View” on page 10-72

Creating Reports Using a Wizard

Application Builder includes many built-in wizards that enable you to generate reports. This section summarizes report options when 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 jQuery Mobile Smartphone).

- Create Application Wizard Report Options
- Create Page Wizard Report Options
- Create Region Wizard Report Options
- Creating a Report Using the Create Application Wizard
- Creating a Report Using the Create Page Wizard
- Creating a Report in Page Designer Using a Wizard

**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 “Building SQL Queries with Query Builder” in Oracle Application Express SQL Workshop Guide.

**See Also:** "Creating a Report in Page Designer Using a Wizard” on page 10-9, "Managing the Application User Interface” on page 7-29, and "Creating a Region in Component View” on page 14-14
### Create Application Wizard Report Options

Table 10–1 describes available report options when using the Create Application Wizard.

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Toggle Report</td>
<td>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. <strong>Note:</strong> Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>List View</td>
<td>Displays the results of a SQL query in a list for mobile applications. <strong>Note:</strong> Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>List View and Form</td>
<td>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. <strong>Note:</strong> Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Reflow Report</td>
<td>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. <strong>Note:</strong> Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Report</td>
<td>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 implementation (Interactive or Classic). <strong>Note:</strong> jQuery Mobile Smartphone only supports classic reports.</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Report and Form</td>
<td>You choose a table on which to build a two page report and form combination. You then choose the report implementation (Interactive or Classic). 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.</td>
<td>Available</td>
<td>Not available</td>
</tr>
</tbody>
</table>

See Also:  
"Creating a Report Using the Create Application Wizard" on page 10-8
Create Page Wizard Report Options

Table 10–2 describes report options when using the Create Page Wizard.

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic Report</td>
<td>Formatted result of a SQL query. You can 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. For assistance in writing the SQL query, click the Query Builder button.</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Understanding the Difference Between Interactive and Classic Reports&quot; on page 10-1 and &quot;Editing Classic Reports in Component View&quot; on page 10-72</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column Toggle Report</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Desktop applications do not support this report format.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive Report</td>
<td>Formatted result of a SQL query. Creates an interactive report based on a custom SQL SELECT statement you provide. For assistance in writing the SQL query, click the Query Builder button. 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.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> jQuery Mobile Smartphone does not support interactive reports.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Understanding the Difference Between Interactive and Classic Reports&quot; on page 10-1, &quot;Customizing Interactive Reports as an End User&quot; on page 10-57, and &quot;Editing Interactive Reports in Component View&quot; on page 10-33</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 10–3 Describe Report Options When Using the Create Region Wizard

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>List View</td>
<td>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. <strong>Note:</strong> Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Reflow Report</td>
<td>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. <strong>Note:</strong> Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Report on Web Service</td>
<td>Creates a report on a Web Service result.</td>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>

**See Also:** "Implementing Web Services" on page 25-7

---

**Create Region Wizard Report Options**

Table 10–3 describes report options when using the Create Region Wizard.
<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive Report</td>
<td>Formatted result of a SQL query. You provide a custom SQL SELECT statement or a PL/SQL function returning a SQL Query statement.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: jQuery Mobile Smartphone does not support interactive reports.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also</strong>: “Customizing Interactive Reports as an End User” on page 10-57, “Editing Interactive Reports in Page Designer” on page 10-11, and “Editing Interactive Reports in Component View” on page 10-33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reflow Report</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Desktop applications do not support this report format.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also</strong>: “Developing Reports” on page 10-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column Toggle Report</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Desktop applications do not support this report format.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classic Report</td>
<td>Formatted result of a SQL query. You provide a custom SQL SELECT statement or a PL/SQL function returning a SQL Query.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also</strong>: “Editing Classic Reports in Page Designer” on page 10-64 and “Editing Classic Reports in Component View” on page 10-72</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ListView</td>
<td>List view is optimized to display data and provide easy navigation on Smartphones.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Desktop applications do not support this report format.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report on Web Service</td>
<td>Creates a report on a Web Service result.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also</strong>: &quot;Implementing Web Services&quot; on page 25-7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**See Also**: “About Regions” on page 14-4
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 Application 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 a name to identify the application.
   c. Application - Enter a unique integer value to identify the application. Application IDs between 3000 to 9000 are reserved for internal use by Oracle Application Express.
   d. User Interface - Select a user interface for the application. For application primarily designed for desktop use, select Desktop.
   e. Theme - Select a theme for this application.
      Themes are collections of templates that you can use to define the layout and style of an entire application. See "Using Themes" on page 13-1.
   f. Theme Style - Select a theme style.
      Theme styles are only available for newer applications such as Universal Theme - 42 and Mobile - 51. See "Using Theme Styles" on page 13-3.
5. Click Next.
6. To add more pages, repeat the previous step.
7. When you are done adding pages, click Next.
8. Follow the on-screen instructions.

Tip: To create an application at this point that consists of just a login page and blank page, click Create Application.

Next, add pages to your application.

5. Click the Add Page button. On Add Page:
      Tip: The Page Type determines if additional attributes display. To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
   b. Parent Page - If this page is subordinate to another page, select the parent page.
   c. Page Name - Specify a name for this page.
6. To add more pages, repeat the previous step.
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 Application Builder icon.
2. Select the application.
3. Click Create Page and click Next.
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.
   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. See "Viewing Field-Level Help" on page 1-21.

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" on page 7-36

Creating a Report in Page Designer Using a Wizard

To add a new report to a page in Page Designer declaratively:

2. Locate the Page Designer Toolbar at the top of the page.
3. On Page Designer Toolbar, click the Create menu and select Create Report Region.
   The Create Region Wizard appears.
4. Select the type of report you wish to create and click Next. To learn more about report options, see "Create Region Wizard Report Options" on page 10-6.
5. Follow the on-screen instructions.

Creating a Report by Dragging and Dropping from the Gallery

To add a report by dragging and dropping from the Gallery:

Page Designer appears.

2. Locate the Gallery at the bottom Grid Layout.

The Gallery lists the types of 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 select a report region and drag it to the appropriate location in Grid Layout.

4. Based on the type of component you add, Page Designer indicates what actions are required next.

   **Tip:** Components can only be dropped into appropriate drop zones, as determined by the component type. If you wish to place the new component next to an existing component then drag the new component to the right, and level, with the existing component, and wait a short period until a new drop zone is displayed.

The Messages tab displays a red or yellow badge indicating the number of messages you need to address. Error messages display in red and Warning messages display in yellow. Selecting an error message highlights the associated attribute in the Property Editor. You must address errors before a page can be saved.

The following example shows a new Interactive Report region with an error message indicating that a SQL statement is required.

5. Edit the appropriate attributes in the Property Editor.

   Required attributes display a red triangle in the upper left corner next above attribute label.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

### 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. This section describes how developers can customize an interactive report in Page Designer and configure how these controls display.

- Editing Region Attributes in Page Designer
- Editing Column Attributes in Page Designer
- Editing Interactive Report Attributes in Page Designer
- Controlling Interactive Report Pagination in Page Designer
- Customizing the Interactive Report Search Bar
- Customizing the Actions Menu in Page Designer
- Managing Link Columns in Page Designer
- Creating a Column Link in an Interactive Report
- Displaying Interactive Reports Conditionally
- Defining a Column as a List of Values in an Interactive Report
- Formatting Columns in an Interactive Report
- Enabling Icon View in Page Designer
- Enabling Detail View in Page Designer
- Configuring Advanced Attributes for Interactive Reports
- Linking to Interactive Reports
- Linking to Shared Interactive Reports

**See Also:** "Managing Pages in Page Designer" on page 8-1, "Customizing Interactive Reports as an End User" on page 10-57, and "APEX_IR" in *Oracle Application Express API Reference*

### Editing Region Attributes in Page Designer

To edit region attributes in Page Designer:

1. View the page containing the region in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.
   Page Designer appears.

2. In the Rendering tree, select the region.
   The Property Editor displays the region attributes in the right column.
Attributes are organized into functional groups. Required attributes display a red triangle in the upper left corner next above attribute label.

3. Edit the appropriate attributes.
   To learn more, see field-level Help.
   
   **Tip:** To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.

4. To save your changes click Save. To save and run the page, click Save and Run Page.

**See Also:** "Property Editor" on page 8-11

### Editing Column Attributes in Page Designer

You can edit Column attributes to precisely control the report layout. For example, you can use these attributes to alter column heading, change column positioning, hide a column, or control how users can manipulate a column.

To access Column attributes:

   Page Designer appears.

2. In the Rendering tree, locate the region and expand the Columns node.
   The columns appear in the Rendering tree.

3. Select a column to edit
   The Property Editor displays the attributes for that column.

4. Edit the appropriate attributes.
To learn more, see field-level Help.

**Tip:** To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.

5. To edit the column heading:
   a. Locate and expand Heading.
   b. Edit the Heading and Alignment attributes.

6. To control how users can manipulate a column:
   a. Locate and expand Enable Users To.
   b. For Enable Users To, select **Yes** or **No** to control how users can manipulate the report.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**See Also:** "Property Editor" on page 8-11

### Editing Interactive Report Attributes in Page Designer

You can edit report attributes to precisely 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 report attributes in Page Designer:

   Page Designer appears.

2. In the Rendering tree, locate the region containing the report.

3. Under Region, select the **Attributes** node.
   The Property Editor displays the report attributes in the Property Editor.

4. Edit the appropriate attributes.
   To learn more, see field-level Help.

   **Tip:** To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

   **See Also:** "**Property Editor"" on page 8-11

### 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:
   Attributes displays in the Property Editor.

2. Configure the 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.

3. To save your changes click Save. To save and run the page, click Save and Run Page.

   Tip: The number of rows you query can impact performance. Querying fewer rows can improve performance and querying thousands of rows can degrade performance. You can control the maximum number of rows to query by editing the Maximum Row Count attribute under Advanced.

Customizing the Interactive Report Search Bar
This section describes how to customize the interactive report search bar.
   ■ Interactive Report Search Bar and Actions Menu
   ■ Customizing the Search Bar in Page Designer
   ■ Removing the Search Bar in Page Designer

   See Also: "Enabling Icon View in Component View" on page 10-48
                  and "Enabling Detail View in Component View" on page 10-49

Interactive Report Search Bar
All interactive reports include 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.

You can remove or customize the search bar by configuring report attributes in the Property Editor.
Customizing the Search Bar in Page Designer

To customize the search bar:


   The Attributes display in the Property Editor.

2. Locate and expand Search Bar.

3. Configure the options under Search Bar. To learn more about an attribute, select the attribute and click the Help tab.

   Table 10–4 describes the Search Bar attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include Search Bar</td>
<td>Select Yes or No. Selecting No removes the Search bar.</td>
</tr>
<tr>
<td>Search Field</td>
<td>Displays a text field in the search bar in which to enter search criteria. Select Yes or No.</td>
</tr>
<tr>
<td>Finder Drop Down</td>
<td>Enables users to search on a selected column or against all columns. Select Yes or No. Within the interactive report, this option is identified as Select columns to search.</td>
</tr>
<tr>
<td>Reports Select List</td>
<td>Displays report select list if the report has alternative default, shared report or personal report. Select Yes or No. See Also: &quot;Saving Interactive Reports&quot; on page 10-60.</td>
</tr>
<tr>
<td>Rows Per Page Selector</td>
<td>Specify whether a separate drop down row selector is included within the search bar. Select Yes or No. See Also: &quot;Saving Interactive Reports&quot; on page 10-60.</td>
</tr>
<tr>
<td>Search Button Label</td>
<td>Specify the text to use as search button label.</td>
</tr>
</tbody>
</table>
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

### Removing the Search Bar in Page Designer

To remove the search bar from in interactive report:

   
   The Attributes page displays in the Property Editor.

2. Locate and expand Search Bar.

3. For Include Search Bar, select **No**.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

### Customizing the Actions Menu in Page Designer

This section describes how to customize the Action menu in an interactive a report.

- **Interactive Report Action Menu**
- **Customizing the Action Menu in Page Designer**
- **Configuring Actions Menu Download Options in Page Designer**

#### Interactive Report Action Menu

The Actions menu appears to the right of the Go button on the Search bar. End users can use this menu to customize an interactive report.

You can remove or customize the Actions menu by configuring report attributes in the Property Editor.
Customizing the Action Menu in Page Designer

To customize the Action menu:


   The Attributes page displays in the Property Editor.

2. Locate and expand Actions Menu.

3. Configure the options under Action Menu by selecting Yes or No.

   **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.

Configuring Actions Menu Download Options in Page Designer

You can configure a Download option to the Actions menu to enable 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.

- **Enabling Download Formats**
- **Enabling CSV Download**
- **Enabling Email Download**

**Enabling Download Formats** To configure download formats:

   
   The Attributes page displays in the Property Editor.

2. View all attributes in the Property Editor. Click the **Show All** button.

3. Edit the following attributes:
   
   To learn more, see field-level Help.
   
   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**. See "Enabling CSV Download" on page 10-18.
      
      - **HTML**.
      
      - **Email**. See "Enabling Email Download" on page 10-19.
      
      - **XLS**.
      
      - **PDF**. See "About Printing Reports to PDF" on page 10-84.
      
      - **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.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**Enabling CSV Download** To enable the Enable CSV download:


   Page Designer appears.

The Attributes page displays in the Property Editor.

4. View all attributes in the Property Editor. Click the **Show All** button.

5. Edit the following attributes:

   To learn more, see field-level Help.

   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:

      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.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**Enabling Email Download** You can configure the Download option to send 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*.


   The Attributes page displays in the Property Editor.

3. Edit the following attributes:

   To learn more, see field-level Help.

   a. Search Bar, Include Search Bar - Select **Yes**.
Editing Interactive Reports in Page Designer

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 **Email**.

   **Tip:** The Download option only appears on the Actions menu if a file format is selected.

4. 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" on page 10-51

Managing Link Columns in Page Designer

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.

- Linking to a Single Row View
- Linking to a Specific Page
- Linking to a URL
- Excluding a Link Column

   **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

To link to a single row view:


   The Attributes page displays in the Property Editor.

2. 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 `<a href= >` 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.

3. 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.

4. 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 make the report accessible. To override this behavior, include meaningful `ALT` text.

### Linking to a Specific Page

To link to a specific page:

   The Attributes page displays in the Property Editor.

2. Under Link:
   a. Link Column - Select **Link to Custom Target**.
   b. 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 Cache - Specify the page numbers on which to clear cache. To specify multiple page, enter a comma-delimited list of page numbers. See "Linking to Interactive Reports" on page 10-28.
      - Request - Specify the request to be used.
      - To reset the pagination for this page, select **Yes** for Reset Pagination.
      - Click **OK**.
   c. Link Icon - Accept the default or specify another image.
   d. Link Attributes (Optional) - Specify additional column link attributes to be included in the `<a href= >` tag (for example, a link target, classes, or styles).
3. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**Linking to a URL**

To link to a URL:


   The Attributes page displays in the Property Editor.

2. Configure the Link attributes:
   
   a. Locate and expand **Link**.
   b. Link Column - Select **Link to Custom Target**.
   c. Configure the Target:
      
      - Click the **Target** field.
      
      The Link Builder - Target dialog appears.
      
      - Target Type - Select **URL**
      
      - URL - Enter the appropriate address.
      
      - 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 `<a href= >` tag (for example, a link target, classes, or styles).

3. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**Excluding a Link Column**

To exclude a column link:


   The Attributes page displays in the Property Editor.

2. Locate and expand **Link**.

3. Configure the Link attributes:
   
   a. Locate and expand **Link**.
   b. For Link Column, select **Exclude Link Column**.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**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:

Page Designer appears.

2. In the Rendering tree, locate the region and expand the Columns node. The columns appear in the Rendering tree.

3. Select a column to edit. The Property Editor displays the attributes for that column.

   **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.

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.
   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**.

**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 region attributes. See "Editing Region Attributes in Page Designer" on page 10-11.

2. Locate and expand Condition.

3. For Type, select a condition type and follow the on-screen instructions.

   Page Designer notifies you of errors and warnings you need to address before you can save the page. The Messages tab displays a red or yellow badge indicating the number of messages you need to address.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.
Tip: Asynchronous JavaScript and XML (Ajax) is used throughout interactive reports. Because Ajax is asynchronous, the value evaluated for conditional display must be in the session. To display an interactive report conditionally based on specified value, use a page item instead of using a :REQUEST value. Using :REQUEST in a conditional display causes the interactive report function incorrectly.

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 region attributes. See "Editing Region Attributes in Page Designer" on page 10-11.
2. Expand the Column node and select the desired column.
3. Locate and expand Column Filter.
4. For Column Filter, 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**.

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 Icon View to an Interactive Report**
- **Adding a Custom Icon View Link**
Adding a View Icon Button
To add Icon view to an interactive report:

   The Attributes page displays in the Property Editor.
2. Locate and expand Icon View.
3. Configure Icon View attributes:

   Tip: To view examples, see field-level Help.

   a. Enabled - Select Yes.

   b. Custom Icon View - Specify whether you wish to define a custom link for the Icon View. Select Yes or No. See "Adding a Custom Icon View Link" on page 10-25.

   c. 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.

   d. Link Column - Select the column that returns the link target of the icon.

   e. Image Source Column - Identify the column that returns the image source.

   f. Label Column - Select the column that returns the image label.

4. To save your changes click Save. To save and run the page, click Save and Run Page.

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:

   The Attributes page displays in the Property Editor.
2. Locate and expand Icon View.
3. Configure Icon View attributes:

   a. Enabled - Select Yes.

   b. Custom Icon View - 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:

   ```html
   <a href="#EMP_LINK#"><img src="#EMP_IMG#" title="#ENAME#" alt="#ENAME#"><br>#ENAME#</a>
   ```

4. To save your changes click Save. To save and run the page, click Save and Run Page.
Formatting Columns in an Interactive Report

You can use HTML expressions in interactive reports to further customize column display.

To define column formatting:
1. Edit the column attributes. See “Editing Column Attributes in Page Designer” on page 10-12.

   The Interactive Report Attributes page appears.
2. Locate and expand Column Formatting.
3. 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
   ```html
   <span style="background-color: yellow; color: red;">#EMPLOYEE_STATUS</span>
   ```

   Example 2: Use Heading 2
   ```html
   <h2>#EMPLOYEE_NAME#</h2>
   ```

   Example 3: Use a dynamic class from the SQL query
   ```html
   <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
   ```html
   <img src="#IMAGE_NAME#" title="#EMPLOYEE_NAME#"> 
   ```

4. To save your changes click Save. To save and run the page, click Save and Run Page.

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:

   The Attributes page displays in the Property Editor.
2. Locate and expand Detail View.
3. Configure Detail View attributes:
   a. Enabled - Select Yes.
   b. Before Rows - Enter the HTML to be displayed before report rows. For example:
c. 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:

\[
\begin{array}{|l|l|}
\hline
\text{ENAME LABEL:} & \text{ENAME} \\
\text{JOB LABEL:} & \text{JOB} \\
\hline
\end{array}
\]

d. After Rows - Enter the HTML to be displayed after report rows. For example:

\[
4. \text{To save your changes click Save. To save and run the page, click Save and Run Page.}
\]

**Configuring Advanced Attributes for Interactive Reports**

This section describes how to configure Advanced interactive report attributes. Advanced attributes enable you to 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
- About Emailing from an Interactive Report

**Configuring Advanced Attributes in Page Designer**

To configure Advanced Attributes:


   The Attributes page displays in the Property Editor.

2. Locate and expand Advanced.

   Table 10–5 describes the available Advanced Attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region Alias</td>
<td>Enter an alphanumeric alias for this interactive report region. The alias must be unique within the application. Enter an alphanumeric alias for this interactive report region. The alias must be unique within the application. You can use an alias to reference an interactive report region in your custom API call. For example, you can write a function to return the INTERACTIVE_REPORT_ID by querying the ALIAS column in the APEX_APPLICATION_PAGE_IR view. Since the INTERACTIVE_REPORT_ID value can change when you export or import the application to different instance, referencing interactive report regions using an alias may be preferable.</td>
</tr>
<tr>
<td>Maximum Row Count</td>
<td>Enter the maximum number of rows to query. The number of rows you query can impact performance. Querying fewer rows can improve performance and querying thousands of rows can degrade performance.</td>
</tr>
<tr>
<td>Report ID Item</td>
<td>Select a page or application item to store the region ID for this report. This enables you to include the region ID, by referencing the value in this item, when defining a URL to navigate to a saved alternative or public report.</td>
</tr>
</tbody>
</table>
To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

### About Emailing from an Interactive Report

You can define the from address for emails sent from an interactive report in two ways:

- At the application-level on the Edit Application Definition page in the Application Email from Address attribute. See "Edit Application Definition Page" on page 7-14.
- At the interactive report-level on the Interactive Report Attributes page in the Email from Address attribute. See "Configuring Advanced Attributes for Interactive Reports" on page 10-27.

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 an email download, but not for subscriptions.

See Also: "Enabling Email Download" on page 10-19 and Understanding Substitution Strings

### 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.
Request Syntax
To link to a specific saved report, use the following Request syntax:

IR[region static ID]_<report_alias>

Consider the following example:

IR[EMP]_dept20

Note: The use of Request in the URL is optional. To link to multiple saved reports, separate each Request with comma.

Name and Value Syntax
To create a filter, use the following itemNames and itemValues syntax:

IR[region static ID]<operator>_<target column alias>

Consider the following example:

IR[EMP]C_ENAME:KING

Note: If there is only one interactive report region on the page, the itemName does not need to be fully qualified with the region static ID. If an operator is not defined, a filter with equal operator is added.

To create a filter on an interactive report in a link, use the itemNames syntax described previously and add the filter value in the corresponding location in the item names section of the URL.

To create row text which contains a filter on an interactive report, use ROWFILTER as the target column alias in the item name syntax and add a filter value in the item itemValues of the URL. The link will add a Row Text Contains filter which searches all columns displayed in the report with type STRING or NUMBER.

Consider the following example:

IR[EMP]_ROWFILTER:SALES
Valid operators include:

- C = Contains
- EQ = Equals (this is the default)
- GTE = Greater than or equal to
- GT = Greater Than
- LIKE = SQL Like operator
- LT = Less than
- LTE = Less than or equal to
- N = Null
- NC = Not Contains
- NEQ = Not Equals
- NLIKE = Not Like
- NN = Not Null
- NIN = Not In (escape the comma separated values with backslash, \\)
- IN = In (escape the comma separated values with backslash, \\)
- ROWFILTER = Row Text Contains (this searches all columns displayed in the report with type STRING or NUMBER)

ClearCache Syntax

**Tip:** Using ClearCache used without specifying a saved report alias in the `Request`, clears the primary default report. To clear a specific interactive report region on a page with more than one interactive reports, specify the region static ID along with saved report alias in the `Request`.

Use the following syntax in the `ClearCache` section of the URL to reset, clear, or reset pagination the interactive report:

- **RIR** - Resets an interactive report. This is equivalent to the end user choosing the Reset option from the interactive report Actions menu on the target page. The report is returned to the default report settings specified by the developer or saved by the user.

- **CIR** - Clears interactive report. This clears all of the report settings such as control break, aggregate, flashback, chart, number of rows to display, filter, highlight, computation, group by, and pivot.

- **RP** - Resets interactive report pagination.

Restrictions When Linking to Interactive Reports

When linking to interactive reports, remember the following restrictions:

1. If a `Request` value is not defined in the URL, all of the primary default report settings will change.
2. If a Request value is not defined and itemNames section does not include a fully qualified filter column name with region static ID, an error occurs if there are more than one interactive report region in the target page.

3. If a Request value is not defined and itemNames section do not include a fully qualified filter column name with region static ID, it will create a filter on the primary default report if there is only one interactive report region in the target page.

Valid Linking Examples
This section contains valid linking examples.

Valid Linking Example 1
This example includes a link with no Request value, but includes a qualified filter column with region static ID in the itemNames section of the URL. This example creates a filter on the primary default report in the referenced region item name.

f?p=&APP_ID.:2:&SESSION.::IR[EMP]_ENAME:KING

Valid Linking Example 2
To support backwards compatibility with previous releases, this example uses older syntax and works on a page with one interactive report region.

?p=&APP_ID.:1:&SESSION.:IR_REPORT_12345::RIR,CIR::RIR,CIR:IR_ENAME:KING

Valid Linking Example 3
In this example the link has no Request value. The clear cache section has RIR or CIR or RP and will reset, clear, or reset the pagination of the primary default reports of all interactive report regions.

f?p=&APP_ID.:1:&SESSION.::RIR:

Invalid Linking Examples
This section contains invalid linking examples.

Invalid Linking Example 1
This link has no Request value and does not include a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error if there is more than one interactive report region in the target page.

f?p=&APP_ID.:2:&SESSION.::ENAME:KING

Invalid Linking Example 2
This example has a link with two request values separated by commas and does not contain a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error since it is not clear where the filter needs to be created.

f?p=&APP_ID.:2:&SESSION.:IR[test]_employee,IR[test2]_dept10::ENAME:KING

Invalid Linking Example 3
This example uses old request value syntax used on a page with multiple interactive report regions. This example will generate an error. The old syntax only works if there is one interactive report region.
Linking to Shared Interactive Reports

This section describes how to link to saved primary default, alternative default, and public reports using \texttt{IR[region static ID]_<report_alias> \ in the Request value of the URL.}

- Editing the Alias of Default Report in Page Designer
- Link Examples to a Primary Report
- Link Examples to Saved Reports

\textbf{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:

2. Under Attributes, expand Saved Reports.
3. Select a saved report.
   The Saved Report attributes display in the Property Editor.
4. Expand Identification.
5. In Alias, enter a new alias for this report. Use this alias to define a more meaningful URL link to this saved report. This link can be used to link directly from elsewhere in the application.
   To learn more, see field-level Help. See “Viewing Help in Page Designer” on page 1-22.
6. To save your changes click \texttt{Save}. To save and run the page, click \texttt{Save and Run Page}.

\textbf{Link Examples to a Primary Report} The following examples demonstrate how to link to a primary report (\texttt{report_alias=PRIMARY}):

This example links to a primary report:
\texttt{f?p=100:1:\&APP\_SESSION.:IR\_PRIMARY}

This example links, resets, and clears primary report settings:
\texttt{f?p=100:1:\&APP\_SESSION.:IR\_PRIMARY::RIR,CIR}

This example links, resets, and clears primary report settings. It additionally creates a \texttt{ENAME = 'KING'} filter on the primary report:
\texttt{f?p=100:1:\&APP\_SESSION.:IR\_PRIMARY::RIR,CIR:IR\_ENAME:KING}

\textbf{Link Examples to Saved Reports} The following examples demonstrate how to link to a saved report (\texttt{report_alias=12345}):

\texttt{f?p=&APP\_ID.:2:&SESSION.:IR\_REPORT\_my rpt:::ENAME:KING}
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::RIR,CIR:IR_ENAME:KING

Editing Interactive Reports in 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. This section describes how developers can customize an interactive report in Component View and configure how these controls display.

- Editing Interactive Report Attributes in Component View
- Editing Column Attributes in Component View
- Creating Column Groups
- Controlling Interactive Report Pagination in Component View
- Managing the Search Bar and Actions Menu in Component View
- Configuring Download Options on the Actions Menu
- Managing Link Columns in Component View
- Creating a Column Link in an Interactive Report
- Displaying Interactive Reports Conditionally
- Defining a Column as a List of Values in an Interactive Report
- Formatting Columns in an Interactive Report
- Enabling Icon View in Component View
- Enabling Detail View in Component View
- Configuring Interactive Report Advanced Attributes
- Linking to Interactive Reports
- Linking to Shared Interactive Reports
- Editing Saved Default Interactive Reports in Component View

See Also:  "Managing Pages in Component View" on page 9-1, "Customizing Interactive Reports as an End User" on page 10-57, and "APEX_IR" in Oracle Application Express API Reference

Editing Interactive Report Attributes in Component View

This section describes how to customize an interactive report by editing report attributes.

- Accessing the Interactive Report Attributes Page in Component View
Report Attribute Page Navigation Alternatives

Accessing the Interactive Report Attributes Page in Component View

To access the Report Attributes page:


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" on page 10-35.
   - Column Groups attributes enable you to group columns on the single row view. See "Creating Column Groups" on page 10-36.
   - 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" on page 10-37.
   - 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" on page 10-41.
   - Link Column attributes enable you to add a link to a column. See "Managing Link Columns in Component View" on page 10-43. and "Creating a Column Link in an Interactive Report" on page 10-45.
   - 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" on page 10-50.

- **Icon View** attributes enable you to add a View Icons icon to the Search bar. See "Enabling Icon View in Component View" on page 10-48.

- **Detail View** attributes enable you to add a View Details icon to the Search bar. See "Enabling Detail View in Component View" on page 10-49.

- **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" on page 1-21.


### 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**.

### 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:


   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 10–6** describes common report column edits.

<table>
<thead>
<tr>
<th>Description</th>
<th>Developer Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change column heading text.</td>
<td>Under Heading, enter different heading text.</td>
</tr>
<tr>
<td>Change column group assignment.</td>
<td>If Column Groups are defined, individual columns can be assigned under Column Group.</td>
</tr>
<tr>
<td>Change how columns display.</td>
<td>Select an option from Display Text As.</td>
</tr>
</tbody>
</table>
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.

### Creating Column Groups

Use column groups to group columns on a single row view.

To create a column group:

   
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**.

### 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:

   
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 10–7 describes the available Pagination attributes.

### Table 10–7 Pagination Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pagination Type</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>■ Row Ranges X to Y</td>
</tr>
<tr>
<td></td>
<td>■ Row Ranges X to Y of Z</td>
</tr>
<tr>
<td>Pagination Display Position</td>
<td>Defines where pagination occurs. Pagination can be display on the left side, right side, at the bottom, or above the report.</td>
</tr>
<tr>
<td>Show Null Values as</td>
<td>Enter the text you want to display for null columns. The default value is (null).</td>
</tr>
<tr>
<td>Maximum Row Count</td>
<td>Defines the maximum number of rows to query, for example, rows 1 - 10 of 456.</td>
</tr>
<tr>
<td></td>
<td>If you set this attribute to 200, the result would appear as follows:</td>
</tr>
<tr>
<td></td>
<td>rows 1 - 10 of more than 200 rows</td>
</tr>
<tr>
<td></td>
<td>This attribute impacts performance. Counting fewer rows can improve performance and counting thousands of rows can degrade performance.</td>
</tr>
<tr>
<td>When more than maximum row data found message</td>
<td>Defines the text message that displays when more than the maximum row count is returned.</td>
</tr>
<tr>
<td>When No Data Found Message</td>
<td>Defines the text message that displays when the query does not return any rows.</td>
</tr>
</tbody>
</table>

3. Click **Apply Changes** to save your edits.

### Managing the Search Bar and Actions Menu in Component View

This section describes how to customize the interactive a report search bar.

- Interactive Report Search Bar and Actions Menu
- Customizing the Search Bar and Actions Menu
- Removing the Search Bar
- Configuring Download Options on the Actions Menu

**See Also:** "Enabling Icon View in Component View" on page 10-48 and "Enabling Detail View in Component View" on page 10-49

### 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.
You can remove or customize the search bar by selecting and deselecting options in the Search Bar section of the Interactive Reports Attributes page.

**Customizing the Search Bar and Actions Menu**

To customize the search bar:


   The Interactive Report Attributes page appears.

2. Scroll down to Search Bar.

   Use these attributes to customize how the search bar displays. Table 10–8 describes Search bar attributes:
### Table 10–8  Search Bar Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include Search Bar</td>
<td>Select <strong>Yes</strong> or <strong>No</strong>. Selecting <strong>No</strong> removes the following default components:</td>
</tr>
<tr>
<td></td>
<td>- Search Field - Displays a text field in the search bar in which to enter search criteria.</td>
</tr>
<tr>
<td></td>
<td>- Finder Drop Down - Enables users to search on a selected column or against all columns.</td>
</tr>
<tr>
<td></td>
<td>- Reports Select List - Displays report select list if the report has alternative default, shared report or personal report. See “Saving Interactive Reports” on page 10-60.</td>
</tr>
<tr>
<td></td>
<td>- Rows Per Page Selector - Displays a select list to enable users to select the number of rows to display per page.</td>
</tr>
<tr>
<td></td>
<td>- Actions Menu - Enables the user to customize the report based on selected Actions.</td>
</tr>
</tbody>
</table>
Table 10–8  (Cont.) Search Bar Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include in Actions Menu</td>
<td>Select and deselect functions to be included in the Actions menu. Available options include:</td>
</tr>
<tr>
<td></td>
<td>■ Select Columns - Used to hide, unhide, and order the report columns.</td>
</tr>
<tr>
<td></td>
<td>■ Filter - Used to add filters to the report (for example, STATE = VA).</td>
</tr>
<tr>
<td></td>
<td>■ Rows Per Page - Used to display rows per page menu.</td>
</tr>
<tr>
<td></td>
<td>■ Sort - Used to sort columns.</td>
</tr>
<tr>
<td></td>
<td>■ Control Break - Used to build control breaks on report columns.</td>
</tr>
<tr>
<td></td>
<td>■ Highlight - Used to define conditions that highlights certain rows.</td>
</tr>
<tr>
<td></td>
<td>■ Compute - Used to add computed columns.</td>
</tr>
<tr>
<td></td>
<td>■ Aggregate - Used to include aggregate calculations on report columns.</td>
</tr>
<tr>
<td></td>
<td>■ Chart - Used to create a Flash chart based off the data in the report.</td>
</tr>
<tr>
<td></td>
<td>■ Group By - Used to create a Group By view based off the data in the report.</td>
</tr>
<tr>
<td></td>
<td>■ Pivot - Used to create a cross tab view based off the data in the report.</td>
</tr>
<tr>
<td></td>
<td>■ Flashback - Used to flashback the report data to some time in the past.</td>
</tr>
<tr>
<td></td>
<td>■ Save Report - Used to save the report settings for future use. Only available for authenticated pages.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ Reset - Used to reset the report to the default settings.</td>
</tr>
<tr>
<td></td>
<td>■ Help - Used to display instructions for using the Interactive report.</td>
</tr>
<tr>
<td></td>
<td>■ Download - Used to download the report in different formats.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>■ Subscription - Used to schedule a subscription to send the report in email. Only available for authenticated pages.</td>
</tr>
</tbody>
</table>

**See Also:** "About the Actions Menu" on page 10-58.
3. Click **Apply Changes**.

**Removing the Search Bar**

To remove the search bar:

2. Scroll down to Search Bar.
3. From Include Search Bar, select **No**.
4. Click **Apply Changes**.

**Configuring Download Options on the Actions Menu**

You can add a Download option to the Actions menu to enable 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.

- **Enabling Download Formats**
- **Enabling CSV Download**
- **Enabling Email Download**

**Enabling Download Formats** To configure download formats:


<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows Per Page</td>
<td>Enter the maximum number to display in the Rows Per Page Selector.</td>
</tr>
</tbody>
</table>
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" on page 10-42.
   - HTML.
   - Email. See "Enabling Email Download" on page 10-43.
   - XLS.
   - PDF. See "About Printing Reports to PDF" on page 10-84.
   - 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.

**Enabling CSV Download** To enable the Enable CSV download:


   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.

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*


   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" on page 10-51

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.

- Linking to a Single Row View
- Linking to a Specific Page
- Linking to a URL
- Excluding a Link Column
Linking to a Single Row View
To link to a single row view:


   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 <a href= > 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 make the report accessible. To override this behavior, include meaningful ALT text.

Linking to a Specific Page
To link to a specific page:


   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 <a href= > 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” on page 10-52.

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.
10. For Name and Value, specify session state for a specific item.
11. Click Apply Changes.

**Linking to a URL**
To link to a URL:

   
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 `<a href= >` 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 user
   
   To learn more about an attribute, see field-level Help.
9. Click **Apply Changes**.

**Excluding a Link Column**
To exclude a column link:

   
The Interactive Report Attributes page appears.
2. Scroll down to Link Column.
3. For Link Column, select **Exclude Link Column**.
4. Click **Apply Changes**.

**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
- Create a Column Link to a URL

**Creating a Column Link to Another Page**
To create a column link to another page:

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 `<a href= >` 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**.

**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**" on page 10-34.

   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 `<a href= >` tag (for example, a link target, classes, or styles).
   d. In URL, enter the appropriate address.
5. Click Apply Changes.

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 Component View" on page 9-5.
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.

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:
   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.

Formatting Columns in an Interactive Report

You can use HTML expressions in interactive reports to further customize column display.

To define column formatting:


   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:

   ```html
   <span style="background-color: yellow; color: red;">#EMPLOYEE_STATUS</span>
   ```

   Example 2:

   ```html
   <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:

   ```html
   <img src="#IMAGE_NAME" title="#EMPLOYEE_NAME">
   ```

5. Click Apply Changes.

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
- Adding a Custom Icon View Link
### Adding Icon View to an Interactive Report

To add Icon view to an interactive report:


   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**.

### Adding a Custom Icon View Link

To add a custom Icon View link:


   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:

   ```html
   <a href='#ICON_LINK#'><img src='#DETAIL_IMG_NO_STYLE#' title='' alt='' width='75' height='75'><br />#PRODUCT_NAME#</a>
   ```

6. Click **Apply Changes**.

### 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:

   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:
     ```html
     <TABLE>
     </TABLE>
     ```
   - For Each Row (Use COLUMN_ALIAS# substitutions) - Use substitution strings for column names and column labels. For example:
     ```html
     <tr><td align="right">#ENAME_LABEL#: #ENAME#</td></tr>
     <tr><td align="right">#JOB_LABEL#: #JOB#</td></tr>
     ```
   - After Rows - Enter the HTML to be displayed after report rows. For example:
     ```html
     </TABLE>
     ```

5. Click Apply Changes.

**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**
- **About Emailing from an Interactive Report**

**Configuring Advanced Attributes**

To configure Advanced Attributes:


   The Interactive Report Attributes page appears.

2. Scroll down to Advanced Attributes.

   Table 10–9 describes the available Advanced Attributes.
3. Click **Apply Changes**.

### 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" on page 7-14.

- **At the interactive report-level** on the Interactive Report Attributes page in the Email from Address attribute. See "Configuring Interactive Report Advanced Attributes" on page 10-50.

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 the &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 an email download, but not for subscriptions.

**Tip:** The email from address comes from the following in the order of precedence:

1. Email From Address at report level
2. Application Email From Address at application level
3. Default Email From Address at instance level

If all of them are null, the email from address is the same as email to address.

**See Also:** "Edit Application Definition Page" on page 7-14
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" on page 10-43 and Understanding Substitution Strings

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.

- Request Syntax
- Name and Value Syntax
- ClearCache Syntax
- Restrictions When Linking to Interactive Reports
- Valid Linking Examples
- Invalid Linking Examples

See Also: "Understanding URL Syntax" on page 2-16 and "About Using f?p Syntax to Link Pages" on page 2-17

Request Syntax

To link to a specific saved report, use the following Request syntax:

IR[region static ID]_<report_alias>

Consider the following example:

IR[EMP]_dept20

Note: The use of Request in the URL is optional. To link to multiple saved reports, separate each Request with comma.

Name and Value Syntax

To create a filter, use the following itemNames and itemValues syntax:

IR[region static ID]<operator><target column alias>

Consider the following example:

IR[EMP]C_ENAME:KING

Note: If there is only one interactive report region on the page, the itemName does not need to be fully qualified with the region static ID. If an operator is not defined, a filter with equal operator is added.
To create a filter on an interactive report in a link, use the `itemNames` syntax described previously and add the filter value in the corresponding location in the item names section of the URL.

To create row text which contains a filter on an interactive report, use `ROWFILTER` as the target column alias in the item name syntax and add a filter value in the item `itemValues` of the URL. The link will add a Row Text Contains filter which searches all columns displayed in the report with type STRING or NUMBER.

Consider the following example:

```
IR[EMP]_ROWFILTER:SALES
```

Valid operators include:

- `C` = Contains
- `EQ` = Equals (this is the default)
- `GTE` = Greater than or equal to
- `GT` = Greater Than
- `LIKE` = SQL Like operator
- `LT` = Less than
- `LTE` = Less than or equal to
- `N` = Null
- `NC` = Not Contains
- `NEQ` = Not Equals
- `NLIKE` = Not Like
- `NN` = Not Null
- `NIN` = Not In (escape the comma separated values with backslash, \)
- `IN` = In (escape the comma separated values with backslash, \)
- `ROWFILTER` = Row Text Contains (this searches all columns displayed in the report with type STRING or NUMBER)

**ClearCache Syntax**

Use the following syntax in the ClearCache section of the URL to reset, clear, or reset pagination the interactive report:

- `RIR` - Resets an interactive report. This is equivalent to the end user choosing the Reset option from the interactive report Actions menu on the target page. The report is returned to the default report settings specified by the developer or saved by the user.
- `CIR` - Clears interactive report. This clears all of the report settings such as control break, aggregate, flashback, chart, number of rows to display, filter, highlight, computation, group by and pivot.
- `RP` - Resets interactive report pagination.

**Restrictions When Linking to Interactive Reports**

When linking to interactive reports, remember the following restrictions:
1. If a Request value is not defined in the URL, all of the primary default report settings will change.

2. If a Request value is not defined and itemNames section does not include a fully qualified filter column name with region static ID, an error occurs if there are more than one interactive report region in the target page.

3. If a Request value is not defined and itemNames section do not include a fully qualified filter column name with region static ID, it will create a filter on the primary default report if there is only one interactive report region in the target page.

Valid Linking Examples
This section contains valid linking examples.

Valid Linking Example 1
This example includes a link with no Request value, but includes a qualified filter column with region static ID in the itemNames section of the URL. This example creates a filter on the primary default report in the referenced region item name.

\[f?p=&APP_ID.:2:&SESSION.::::IR[EMP]_ENAME:KING\]

Valid Linking Example 2
To support backwards compatibility with previous releases, this example uses older syntax and works on a page with one interactive report region.

\[?p=&APP_ID.:1:&SESSION.:IR_REPORT_12345::RIR,CIR::RIR,CIR:IR_ENAME:KING\]

Valid Linking Example 3
In this example the link has no Request value. The clear cache section and has RIR or CIR or RP and will reset, clear, or reset the pagination of the primary default reports of all interactive report regions.

\[f?p=&APP_ID.:1:&SESSION.:::RIR:\]

Invalid Linking Examples
This section contains invalid linking examples.

Invalid Linking Example 1
This link has no Request value and does not include a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error if there is more than one interactive report region in the target page.

\[f?p=&APP_ID.:2:&SESSION.::::ENAME:KING\]

Invalid Linking Example 2
This example has a link with two request values separated by commas and does not contain a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error since it is not clear where the filter needs to be created.

\[f?p=&APP_ID.:2:&SESSION.:IR[test]_employee,IR[test2]_dept10::ENAME:KING\]
Invalid Linking Example 3
This example uses old request value syntax used on a page with multiple interactive report regions. This example will generate an error. The old syntax only works if there is one interactive report region.

```
f?p=&APP_ID.:2:&SESSION.:IR_REPORT_myrpt:::ENAME:KING
```

Linking to Shared Interactive Reports
This section describes how to link to saved primary default, alternative default, and public reports using `IR_REPORT_[report_alias]` in the request value of the URL.

- Link Examples to a Primary Report
- Link Examples to Saved Reports

**Tip:** You can edit the report alias or view a link example on the Saved Reports page. See "Editing the Alias of Default Report in Component View" on page 10-56.

Link Examples to a Primary Report
The following examples demonstrate how to link to a primary report (report_alias=12345):

- This example links to a primary report:
  
  ```
  f?p=100:1:&APP_SESSION.:IR_REPORT_12345
  ```

- This example links, resets, and clears primary report settings:
  
  ```
  f?p=100:1:&APP_SESSION.:IR_REPORT_12345::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_REPORT_12345::RIR,CIR::IR_ENAME:KING
  ```

Link Examples to Saved Reports
The following examples demonstrate how to link to a saved report (report_alias=12345):

- This example links to a saved report:
  
  ```
  f?p=100:1:&SESSION.:IR_REPORT_1234
  ```

- This example links, resets, and clears settings for saved report 12345:
  
  ```
  f?p=100:1:&SESSION.:IR_REPORT_12345::RIR,CIR:
  ```

- This example links, resets, and clears saved report 12345 settings. It additionally creates a `ENAME = 'KING'` filter on saved report 12345.
  
  ```
  f?p=100:1:&SESSION.:IR_REPORT_12345::RIR,CIR::RIR,CIR:IR_ENAME:KING
  ```

If you reset, clear, or create filter using URL without the `IR_REPORT_[report_alias]` request value, it resets, clears, and creates filter on the primary report. For example, this link goes to the primary report:

```
f?p=100:1:&SESSION:::RIR,CIR:IR_ENAME:KING
```
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
- Deleting Saved Default Reports in Component View

See Also: "Saving a Default Interactive Report" on page 10-63 and "Managing Saved Interactive Reports" in Oracle Application Express Administration Guide

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 Component View" on page 9-5.
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.

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 Component View" on page 9-5.
2. Under Regions, click the Saved Report link.
   The Saved Reports page appears.
3. To delete a report, select it and click Delete Checked.

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:

   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, Monitor Activity and By Interactive Report.

Customizing Interactive Reports as an End User

Users can alter the layout of interactive reports data by choosing the columns they are interested in, applying filters, highlighting, and sorting. They can also define breaks, aggregations, charts, group bys, and add their own computations.

- About the Search Bar
- About the Actions Menu
- Using Interactive Report Filters
- Saving Interactive Reports

See Also: "Creating Reports Using a Wizard" on page 10-3 and "Customizing an Interactive Report Using the Actions Menu" in Oracle Application Express End User Guide

About the Search Bar

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" on page 10-60.
- 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" on page 10-58.
About the Actions Menu

You can customize an interactive report by selecting options on the Actions menu.

**Tip:** Not all options described in this section are available on every Actions menu. Developers can customize what options appear. See "Customizing the Actions Menu in Page Designer" on page 10-16 and "Customizing the Search Bar and Actions Menu" on page 10-38.

The Actions menu appears to the right of the Go button on the Search bar. Use this menu to customize an interactive report.

The Actions menu contains the following options:

- **Select Columns** specifies which columns to display and in what order.
- **Filter** focuses the report by adding or modifying the `WHERE` clause on the query.
- **Rows Per Page** determines how many rows display in the current report.
- **Format** contains the following submenu:
  - **Sort** - Changes the columns to sort on and determines whether to sort in ascending or descending order.
  - **Control Break** - Creates a break group on one or several columns. This pulls the columns out of the interactive report and displays them as a master record.
  - **Highlight** - Defines a filter that highlights the rows that meet the filter criteria.
  - **Compute** - Enables users to add computed columns to a report.
  - **Aggregate** - Enables users to perform mathematical computations against a column.
  - **Chart** - Displays the report data as a chart.
  - **Group By** - Enables users to group the result set by one or more columns and perform mathematical computations against columns.
– **Pivot** - Enables users to define a Pivot report.

- **Flashback** - Enables you to view the data as it existed at a previous point in time.
- **Save Report** - Saves the interactive report. Depending upon their user credentials, users can save different types of reports.
- **Reset** - Resets the report back to the default report settings.
- **Help** - Provides descriptions of how to customize interactive reports.
- **Download** - Enables users to download a report. Available download formats depend upon your installation and report definition.
- **Subscription** - Enables users to send themselves an interactive report by email.

**See Also:** "Customizing an Interactive Report Using the Actions Menu" in Oracle Application Express End User Guide

### Using Interactive Report Filters

When you customize an interactive report, a filter displays between the search bar and the report. The following illustration shows a report with two filters `Project = ‘Bug Tracker’` and `Status = ‘On-Hold’`.

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 Project and Status filter details hidden.

Other keys features of interactive report filters include:

- **Enable/Disable** - To enable and disable a filter, select and deselect the checkbox 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.

Saving Interactive Reports
This section describes how to save an interactive report. All users can save a private or public interactive report. However, only the user who creates a private report can view, save, rename, or delete it. Developers have additional save capabilities in that they can save the report that initially displays, called the Primary Default, or create an Alternative Default report.

About the User Type and Available Save Options
About Configuration Dependencies
Saving a Public or Private Interactive Report
Saving a Default Interactive Report
Renaming a Public or Private Interactive Report
Deleting a Public or Private Interactive Report

See Also: "Managing Saved Interactive Reports" in Oracle Application Express Administration Guide

About the User Type and Available Save Options
An end user can save an interactive report in two ways:

Public Report (End user and developer). This report is viewable by all users. However, only the user who creates a public report can save, rename, or delete it. Although all users can view a public report, they can only save it under a new report name.

Private Report (End user and developer). Only the user who creates the private report can view, save, rename, or delete it.

A developer can additionally save an interactive report in the following ways:

Primary Default (Developer only). The Primary Default is the report that initially displays. Primary Default reports cannot be renamed or deleted.

Alternative Report (Developer only). Enables developers to create multiple report layouts. Only developers can save, rename, or delete an Alternative Report.

The following shows the Reports list on the Search bar of an interactive report.
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, Monitor Activity and By Interactive Report.

### About Configuration Dependencies

The Save Report option on the Actions only displays if the application developer:


Also note that the Reports select list on the Search Bar only displays if the application developer selects Reports Select List under Search Bar on the Report Attributes page. See "Managing the Search Bar and Actions Menu in Component View" on page 10-37.

Note that many of these options are selected by default.

### 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 select Save Public Report under Include in Actions Menu on the Report Attributes page. Developers can also apply authorization scheme to enable or disable Save Public Report. See "Managing the Search Bar and Actions Menu in Component View" on page 10-37.

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). See "About the Actions Menu" on page 10-58.

3. Click the Actions menu and select Save Report.

The Save Report dialog appears.

4. For end users, the following Save Report dialog appears.

   ![Save Report Dialog](image)

   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.

   ![Save Report Dialog](image)

   In Save Report:

   a. Save - Select As Named Report.
   b. Name - Enter a name for the report.
   c. Public - Select this check box to make the report viewable to all users. Deselect this check box to make the report private.

   **Tip:** The Public check box only displays if you have enabled the Save Public Report report attribute.

   d. Description - Enter an optional description.
   e. Click Apply.
Saving a Default Interactive Report
Developers can save two types of default interactive report: primary and alternative. Both reports display on the Report list on the search bar. However, primary default reports cannot be renamed or deleted.

To save a default interactive report:
1. Run the report as a developer.
2. Customize the report (for example, hide columns, add filters, and so on). See “About the Actions Menu” on page 10-58.
3. Click the Actions menu and select Save Report.
   The Save Report dialog appears.

4. From Save, select As Default Report Settings.
   The Save Default Report dialog appears.

5. For Default Report Type, select either:
   - Primary - This report displays as the primary report when the page runs.
   - Alternative - This report displays as an alternative option on the Reports list on the Search bar. If you select Alternative, enter a name.

6. Click Apply.

Renaming a Public or Private Interactive Report
To rename a public or private interactive report:
1. Run the report as a developer.
2. Select public or private interactive report to rename.
3. Click the saved report link.
4. In the Rename Report dialog, edit the attributes (for example, enter a new name) and click **Apply**.

**Deleting a Public or Private Interactive Report**

To delete a public or private interactive report:

1. Run the report as a developer.
2. Click the **Remove Report** icon next to the report name link.
3. Click **Apply**.

**About Exporting Interactive Reports**

When you export an application, the Primary Default, Alternative Default, public reports are included by default. You can also choose to include personal reports during the export process by selecting **Yes** for Private Interactive Reports. See "Exporting an Application" on page 24-14.

**Editing Classic Reports in Page Designer**

This section describes how developers can edit classic reports. Using the Report Attributes page, developers can control report layout, pagination, column sorting, error messages, export links, and column break of Classic reports.
Tip: With the addition of interactive reports in Oracle Application Express release 3.1, all previous report types are referred to as classic reports.

See Also: "Managing Pages in Page Designer" on page 8-1 and "APEX_IR" in Oracle Application Express API Reference

- Editing Classic Report Attributes in Page Designer
- Altering Classic Report Layout in Page Designer
- Controlling Classic Report Pagination in Page Designer
- Enabling Column Sorting in a Classic Report in Page Designer
- Adding a Download Link to a Classic Report in Page Designer
- Enabling Classic Report CSV Output Option in Page Designer
- Adding Support for XML File or a CSV File Export in Page Designer
- Creating a Column Link in a Classic Report in Page Designer
- Defining an Updatable Column in a Classic Report in Page Designer
- Defining a Column as a List of Values in a Classic Report in Page Designer
- Controlling When Classic Report Columns Display in Page Designer
- Controlling Classic Report Column Breaks in Page Designer

Editing Classic Report Attributes in Page Designer

This section describes how to customize a classic report by editing report attributes in Page Designer.

To edit report attributes:

   Page Designer appears.
2. In the Rendering tree, expand the region containing the report and select the Attributes node.
The Property Editor displays the report attributes. Attributes are organized into functional groups. Required attributes display a red triangle in the upper left corner next above attribute label.

3. Edit the appropriate attributes.

   **Tip:** To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.

4. 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**.

**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:


   Page Designer appears.

2. In the Rendering tree, locate the region and expand the Columns node.

   The columns appear in the Rendering tree.

3. Select a column to edit.

   The Property Editor displays the attributes for that column.

4. Edit the appropriate attributes.

   **Tip:** To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.

   **Table 10–10** describes common report column edits.

<table>
<thead>
<tr>
<th>Description</th>
<th>Developer Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide a column.</td>
<td>Under Type, select <strong>Hidden Column</strong>.</td>
</tr>
<tr>
<td>Edit the column heading text.</td>
<td>Under Heading, edit <strong>Heading</strong> and <strong>Alignment</strong>.</td>
</tr>
<tr>
<td>Change the column heading text</td>
<td>Under Heading, make a new selection from <strong>Alignment</strong>.</td>
</tr>
<tr>
<td>alignment.</td>
<td></td>
</tr>
<tr>
<td>Alter the column display sequence.</td>
<td>Under Layout, edit <strong>Sequence</strong>.</td>
</tr>
</tbody>
</table>
| Enable a unique sort sequence.     | Under Sorting, for Sortable click **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.
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**.

### Controlling Classic Report Pagination in Page Designer

This section describes how to control pagination in a classic report.

- **About Controlling Pagination in a Classic Report**
- **Editing Classic Report Pagination Attributes**

#### About Controlling 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 Pagination attributes on the Report Attributes page.

#### Editing Classic Report Pagination Attributes

To control classic report pagination:


   The Property Editor displays the report attributes in the right column.

2. Locate and expand Pagination.

3. Under Pagination, edit the following attributes:

   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.

4. 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**.
Enabling Column Sorting in a Classic Report in Page Designer

You enable column sorting on the Report Attributes page.

To enable column sorting:


   Page Designer appears.

2. In the Rendering tree, locate the region and expand the Columns node.

   The columns appear in the Rendering tree.

3. Select a column to edit

   The Property Editor displays the attributes for that column.

   **Tip:** To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.

4. Locate and expand Sorting. Edit the following attributes:

   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.

5. Specify ascending and descending image attributes or click **set defaults**.

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" on page 10-75.

Adding a Download Link to a Classic Report in Page Designer

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" on page 7-29

Enabling Classic Report CSV Output Option in Page Designer

To use the enable the CSV output option:


   The Property Editor displays the report attributes in the right column.

2. Locate and expand Download.

   **Tip:** To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.
3. Under Download, edit the following attributes:
   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.

4. To save your changes click Save. To save and run the page, click Save and Run Page.

Adding Support for XML File or a CSV File Export 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:

   The Property Editor displays the report attributes in the right column.
2. Locate and expand Appearance.
   Tip: To control the amount of information that displays, click the Show Common, Show All, Collapse All, and Expand All buttons.

3. 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.
4. To save your changes click Save. To save and run the page, click Save and Run Page.

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:

   Page Designer appears.
2. In the Rendering tree, locate the region and expand the Columns node.
   The columns appear in the Rendering tree.
3. Select a column to edit
   The Property Editor displays the attributes for that column.
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. 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 a URL, in the Link Builder - Target dialog:
   a. Type - Select URL.
   b. URL - Enter the URL address.
   c. Click OK.
8. To save your changes click Save. To save and run the page, click Save and Run Page.

Defining an Updatable Column in a Classic Report in Page Designer
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:
   Page Designer appears.
2. In the Rendering tree:
   a. Locate the region containing the report or tabular form.
   b. Expand the Columns node.
   c. Select the column to be updated.
3. In the Property Editor, expand Identification and edit the Type attribute.
   To learn more, see field-level Help.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

### Defining a Column as a List of Values in a Classic Report 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:

   Page Designer appears.
2. In the Rendering tree:
   a. Locate the region containing the report or tabular form.
   b. Expand the Columns node.
   c. Select the column to be updated.
3. In the Property Editor, expand Identification and change the Type to **Plain Text (based on List of Values)**.
4. Expand List of Values and select a Type and fill in the appropriate fields.
   To learn more, see field-level Help.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

### 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:

   Page Designer appears.
2. In the Rendering tree, 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 Conditions.
   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" on page 20-49, "Understanding Conditional Rendering and Processing" on page 2-12, and Appendix C, "Available Conditions" on page C-1

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:

   Page Designer appears.

2. In the Rendering tree, locate the region and expand the Columns node.
   The columns appear in the Rendering tree.

3. Select a column to edit.
   The Property Editor displays the attributes for that column.

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.

Editing Classic Reports in Component View

This section describes how developers can 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.
Tip: With the addition of interactive reports in Oracle Application Express release 3.1, all previous report types are referred to as classic reports.

See Also: "Managing Pages in Component View" on page 9-1, "Customizing Interactive Reports as an End User" on page 10-57, and "APEX_IR" in Oracle Application Express API Reference

- Editing Classic Report Attributes in Component View
- Altering Classic Report Layout in Component View
- Controlling Classic Report Pagination in Component View
- Enabling Column Sorting in a Classic Report in Component View
- Adding a Download Link to a Classic Report in Component View
- Enabling Classic Report CSV Output in Component View
- Adding Support for XML File or a CSV File Export
- Creating a Column Link in a Classic Report in Component View
- Defining an Updatable Column in Component View
- Defining a Column as a List of Values in a Classic Report in Component View
- Controlling When Classic Report Columns Display in Component View
- Controlling Classic Report Column Breaks in Component View
- Synchronizing Classic Report Column Order with Region Source

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:


   Component View appears.

2. To access the Report Attributes page, click the Report link next to the report region you want to edit.

   ![Regions and Report](image)

   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" on page 10-74.

- **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 a 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" on page 10-78 and "Adding Support for XML File or a CSV File Export" on page 10-78.

- **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" on page 10-82.

- **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" on page 1-21.

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

**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:


   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 10–11 describes common report column edits.

<table>
<thead>
<tr>
<th>Description</th>
<th>Developer Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alter column display sequence.</td>
<td>Click the up and down arrows to change the column display sequence.</td>
</tr>
<tr>
<td>Alter heading alignment.</td>
<td>Under Column Alignment, select a column alignment.</td>
</tr>
<tr>
<td>Change column heading text.</td>
<td>Under Heading, enter different heading text.</td>
</tr>
<tr>
<td>Control which columns display.</td>
<td>Click <strong>Show</strong> to indicate a column should display.</td>
</tr>
<tr>
<td>Enable a unique sort sequence.</td>
<td>Click <strong>Sort</strong> and select a sequence number from <strong>Sort Sequence</strong>.</td>
</tr>
<tr>
<td></td>
<td>Any number of columns can be sort enabled. However, at least one column must have a Sort Sequence defined.</td>
</tr>
<tr>
<td>Enable the sum of a column.</td>
<td>Click <strong>Sum</strong> to enable the sum of a column.</td>
</tr>
</tbody>
</table>
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 Alias.

3. To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.


### Controlling Classic Report Pagination in Component View

This section describes how to control pagination in a classic report.

- **About Controlling Pagination in a Classic Report**
- **Controlling Classic Report Pagination**
- **Including Pagination After the Rows in a Classic Report**
- **Including Pagination Before the Rows in a Classic Report**

#### About Controlling 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.

#### Controlling Classic Report Pagination

To control classic report pagination:

   
   The Report Attributes page appears.

2. Under Layout and Pagination, select the following:
   
   
   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 10–12.

   To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
Including Pagination After the Rows in a Classic Report

To include pagination after the rows in a report:


   The Report Attributes page appears.

3. Under Layout and Pagination, specify the following:
   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:
   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.

Including Pagination Before the Rows in a Classic Report
To include pagination before the rows in a report:


   The Report Attributes page appears.

3. Under Layout and Pagination:
   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.
   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.

Enabling Column Sorting in a Classic Report in Component View
You enable column sorting on the Report Attributes page.
To enable column sorting:
   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.

---

**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" on page 7-29

---

**Enabling Classic Report CSV Output in Component View**

To use the enable the CSV output option:

   The Report Attributes page appears.


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.

---

**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:


   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**.

---

**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
- Creating a Column Link to a URL

**Creating a Column Link to Another Page**

To create a column link to another page:


   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 `<a href= >` 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" on page 10-28.

   g. Use the Name and Value fields to specify session state for a specific item.

6. Click **Apply Changes**.

**Creating a Column Link to a URL**

To create a column link to a URL:

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 <a href= > tag (for example, a link target, classes, or styles).
   d. In URL, enter the appropriate address.

5. Click Apply Changes.

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:


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.

**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:

   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.

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:

   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" on page 20-49,  "Understanding Conditional Rendering and Processing" on page 2-12, and Appendix C, "Available Conditions" on page C-1

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:

   The Report Attributes page appears.
2. Scroll down to Break Formatting.
3. Make a selection from the Breaks list.

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:

   The Report Attributes page appears.
2. From Tasks, select Synchronize Column Order with Region Source.
3. Click Synchronize.

Printing Report Regions

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
- About Report Printing Configuration Options
- About Classic Report Printing Methods
- Configuring Classic Report Region Print Attributes
- Configuring Interactive Report Region Print Attributes
- About Report Queries
- About Report Layouts

**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” on page 20-10.
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, they simply click a print link. However, from developer’s perspective, you must then declaratively create regions to support PDF printing.

About Report Printing Configuration Options

Oracle Application Express provides three report printing configuration options:

- **Oracle REST Data Services** - Select this option if you are using the Oracle REST Data Services (formerly called Application Express Listener) release 2.0 or later. This option enables you to use the basic printing functionality, which includes creating report queries and printing report regions using the default templates provided in Application Express and using your own customized XSL-FO templates.

  **Note:** The Oracle REST Data Services option does not require an external print server, instead the report data and style sheet are downloaded to the listener, rendered into PDF format by the listener and then sent to the client. The PDF documents in this setup are not returned back into the database, thus the print APIs are not supported when using the Oracle REST Data Services-based configuration.

- **External (Apache FOP)** - Select this option if you are using Apache FOP on an external J2EE server. This option enables you to use the basic printing functionality, which includes creating report queries and printing report regions using the default templates provided in Application Express and using your own customized XSL-FO templates.

- **Oracle BI Publisher** - This option requires a valid license of Oracle BI Publisher (also known as Oracle XML Publisher). This option enables you to take report query results and convert them from XML to RTF format using Oracle BI Publisher. Select this option to upload your own customized RTF or XSL-FO templates for printing reports within Application Express.

  **See Also:** *PDF Printing in Application Express* to learn more about installing and configuring Oracle BI Publisher. Go to:


To use the full functionality of report printing, your Oracle Application Express service administrator must enable it for your instance. To learn more, see "Configuring Report Printing" in Oracle Application Express Administration Guide.

About Classic Report Printing Methods

There are two ways to print classic report regions:

- **Configure Printing Attributes for a Report Region.** You can print a report region by configuring the Printing Attributes for the region. See "Configuring Classic Report Region Print Attributes" on page 10-85.

- **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" on
Both report regions and report queries can be downloaded in the following formats:

- **PDF** - Adobe Portable Document 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. To learn more, see "Formatting a Report Region or Report Query Using Report Layouts" on page 18-41.

**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 HTM (to download the document as an HTML file).

**Configuring Classic Report Region Print Attributes**

One approach to printing a report region is to configure Print Attributes. Once configured, these attributes only apply only to 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**
- **Configuring a Classic Report Print Attributes in Component View**

**See Also:** "About Classic Report Printing Methods" on page 10-84 and "Configuring Report Printing" in Oracle Application Express Administration Guide

**Configuring a Classic Report Print Attributes in Page Designer**

To configure a classic report region for printing:

   
   Page Designer appears.

2. Enable the Printing attribute:

   a. View the page containing the classic report in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.

   b. In the Rendering tree, expand the region containing the report and select the Attributes node
      
      The Property Editor displays the report attributes.

   c. In the Property Editor, locate expand the Printing section.

   d. For Printing, Enabled, select Yes.

   e. Click **Save** and refresh the page.
3. In the Rendering tree, expand the Attributes node and select the **Printing** node.

4. Under Region, select the **Attributes** node.

5. Edit the appropriate Printing attributes.
   
   To learn more, see field-level Help.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

**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.

   h. Report Layout - Select **Default Report Layout** to use the built-in generic report layout, or an available report layout you or your administrator created.

      For a listing of the supported substitution strings available when building custom XSL-FO report layouts, see field-level Help.

   i. 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 makes debugging and diagnosis of potential printing problems easier and supports the use of specific custom built end points to perform custom XSLT transformations.

4. Print URL - Report regions can be printed using the standard print link or using a page button. To use a button, remove the Link Label, create a button on your page, and use the URL shown in this attribute as the button target.

5. Under Report Columns, specify the width for each column.

   The default is to make each column the same width. The Total Width is a display only field that shows the sum of all column width values. When changing the width for an individual column the Total Width is automatically recalculated. If Column Width Units is set to Percentage, rather than Points, then the total column width must be less than or equal to 100.

   Use the Reset Column Width button to reset the Column Width Units to Percentage and change all column widths back to their default value. The Recalculate button maintains the current ratios between column widths but reduce all column widths such that the Total Width is less than or equal to 100%.

6. Click Apply Changes.

Configuring Interactive Report Region Print Attributes

Unlike classic reports, the Interactive Report Print Attributes can only use the default XSL-FO layout and is initiated from the Report Attributes, Download section rather than directly from this screen. Once configured, these attributes only apply only to the current region and cannot be used outside the context of the region.

If the printing feature is set up for your instance, you can configure a report region to print in various formats.

- Configuring Interactive Report Regions for Printing in Page Designer
- Configuring Interactive Report Regions for Printing in Component View

Configuring Interactive Report Regions for Printing in Page Designer

To configure an interactive report region for printing:


   Page Designer appears.

2. In the Rendering tree, expand the region containing the report and locate the Attributes node.

3. Expand the Attributes node and select the Printing node.

   The Printing Attributes display in the Property Editor.

4. Edit the appropriate Printing attributes.

   To learn more, see field-level Help.

5. To save your changes click Save. To save and run the page, click Save and Run Page.
Configuring Interactive Report Regions for Printing in 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.

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" on page 18-38

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.
Understanding BLOB Support in Forms and Reports

Oracle Application Express includes declarative BLOB support to enable developers to declaratively upload files in forms, and download or display files in reports. BLOB display and download can also be authored procedurally using PL/SQL. This section describes how to upload, download and display files, and how to manage additional file attributes such as MIME type and file name that are important for proper management of files stored in BLOB columns. Using this functionality you can easily extend your Oracle Application Express applications to manage files including images, documents, videos, and so on.

- About BLOB Support in Reports
- About Providing a Download Link
- About Displaying the BLOB
- About Working With BLOBs Procedurally

About BLOB Support in Reports

Oracle Application Express includes BLOB support for both classic and interactive reports. If you use a wizard to create a report and include a column of type BLOB, basic support is included. Additional information should be added after generation to make the download capability more user friendly.

About Providing a Download Link

To facilitate the inclusion of a download link in a report, the report includes the selection of the length of the BLOB (for example, dbms_lob.getlength(RESUME)). If the length is 0, the BLOB is NULL and no download link is displayed. In the same way you specify a format mask for a date or number you can format a download link. The DOWNLOAD format is more complex however then 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:

<table>
<thead>
<tr>
<th>Position</th>
<th>Attribute</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DOWNLOAD</td>
<td>Yes</td>
<td>Identifies the DOWNLOAD report format mask.</td>
</tr>
<tr>
<td>2</td>
<td>Table Name</td>
<td>Yes</td>
<td>Case sensitive name of table containing target column of type BLOB.</td>
</tr>
<tr>
<td>3</td>
<td>Column containing BLOB</td>
<td>Yes</td>
<td>Case sensitive name of column of type BLOB.</td>
</tr>
<tr>
<td>4</td>
<td>Primary Key Column 1</td>
<td>Yes</td>
<td>Case sensitive name of primary key column 1.</td>
</tr>
<tr>
<td>5</td>
<td>Primary Key Column 2</td>
<td>No</td>
<td>Case sensitive name of primary key column 2.</td>
</tr>
<tr>
<td>6</td>
<td>MIME type Column</td>
<td>No</td>
<td>Case sensitive column name used to store the MIME type.</td>
</tr>
<tr>
<td>7</td>
<td>Filename Column</td>
<td>No</td>
<td>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.</td>
</tr>
</tbody>
</table>
Consider the following example:

DOWNLOAD:EMP:RESUME:EMPNO::RESUME_MIMETYPE:RESUME_FILENAME:RESUME_LAST_UPDATE::attachment:Resume

To be able to enter these parameters and create the format as described above, you have to select **Blob Format** for Number/Date format item. Once selected, the Blob Column attributes region displays and you can fill in all parameters as described above.

### About Displaying the BLOB

If the BLOB you are working with is an image, you can display it in the report using the new report format mask of 'IMAGE'. Regardless of the **MIME** type, the report always attempts to display the BLOB. If the BLOB cannot be rendered, a broken image is displayed.

The following table describes the parameters of the IMAGE format mask:

<table>
<thead>
<tr>
<th>Position</th>
<th>Attribute</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IMAGE</td>
<td>Yes</td>
<td>Identifies the IMAGE report format mask.</td>
</tr>
<tr>
<td>2</td>
<td>Table Name</td>
<td>Yes</td>
<td>Case sensitive name of table containing target column of type BLOB.</td>
</tr>
<tr>
<td>3</td>
<td>Column containing BLOB</td>
<td>Yes</td>
<td>Case sensitive name of column of type BLOB.</td>
</tr>
<tr>
<td>4</td>
<td>Primary Key Column 1</td>
<td>Yes</td>
<td>Case sensitive name of primary key column 1.</td>
</tr>
<tr>
<td>5</td>
<td>Primary Key Column 2</td>
<td>No</td>
<td>Case sensitive name of primary key column 2.</td>
</tr>
<tr>
<td>6</td>
<td>MIME type Column</td>
<td>No</td>
<td>Case sensitive column name used to store the MIME type.</td>
</tr>
</tbody>
</table>
Consider the following example:

```plaintext
IMAGE:EMP:RESUME:EMPNO::RESUME_MIMETYPE:RESUME_FILENAME:RESUME_LAST_UPDATE::attachment:Resume
```

To be able to enter these parameters and create the format as described above. You have to select Blob Format for Number/Date format item. Once selected, a new region, Blob Column attributes, displays and you can fill in all parameters as described above.

### About Working With BLOBs Procedurally

As an alternative to using the built-in methods of providing a download link, you can use the `APEX_UTIL.GET_BLOB_FILE_SRC` function. One advantage of this approach, is the ability to more specifically format the display of the image (with height and width tags). Note that this approach is only valid if called from a valid Oracle Application Express session. Also, this method requires that the parameters that describe the BLOB to be listed as the format of a valid item within the application. That item is then referenced by the function.

**See Also:** "GET_BLOB_FILE_SRC Function" in the *Oracle Application Express API Reference*
You can include a variety of different types of forms in your applications. You can include forms that enable users to update a single row in a table or multiple rows. This section describes how to use wizards to create forms, how to create forms manually, and how to validate user input.

- Using a Wizard to Create a Form
- Managing Tabular Forms
- Managing Master Detail Forms
- Creating a Form Manually in Component View
- Validating User Input in Forms
- Understanding BLOB Support in Forms

See Also: "Developing Reports" on page 10-1

Using a Wizard to Create a Form

Application Builder includes built-in wizards that enable you to generate forms. This section summarizes form options when using 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 and jQuery Mobile Smartphone).

- Create Page Wizard Form Options
- Create Region Wizard Form Options
- Creating a Form Using the Create Page Wizard
- Creating a Form Using the Create Application Wizard
- Creating a Form in Page Designer

See Also: "Creating a Report Using the Create Application Wizard" on page 10-8, "Creating a Report Using the Create Page Wizard" on page 10-9, and "Creating a Region in Component View" on page 14-14

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 "Building SQL Queries with Query Builder" in Oracle Application Express SQL Workshop Guide.
Create Page Wizard Form Options

Table 11–1 describes form options when using the Create Page Wizard. Note that the available forms options differ depending the selected user interface (for example, Desktop and jQuery Mobile Smartphone).

Table 11–1  Create Page Wizard Report Options

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form on a Procedure</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Form on a Table or View</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Form on a Table with List View</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Form on a Table with Report</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td>Note: This wizard does not support tables having more than 127 columns. Selecting more than 127 columns generates an error.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Master Detail Form</td>
<td>Creates a form that displays a master row and multiple detail rows within a single HTML form. With this form, users can query, insert, update, and delete values from two tables or views. You choose the tables on which to build the master report and detail form.</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>See Also: “Managing Master Detail Forms” on page 11-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tabular Form</td>
<td>Creates a form to perform update, insert, and delete operations on multiple rows in a database table. You choose the table on which to build the form.</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>See Also: “Managing Tabular Forms” on page 11-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form on a SQL Query</td>
<td>Creates a form based on the columns returned by a SQL query such as an EQUILJOIN.</td>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>
Table 11–2  Create Region Wizard Form Options

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary Page</td>
<td>Creates a read-only version of a form. Use this option to create a confirmation page at the end of a wizard.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form on Web Service</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating a Form on a Web Service&quot; on page 25-17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form and Report on Web Service</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating an Input Form and Report on a Web Service&quot; on page 25-15</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See Also: "Creating a Form Using the Create Page Wizard" on page 11-4

Create Region Wizard Form Options

Table 11–2 describes form options when using the Create Region Wizard.

Table 11–2  Create Region Wizard Form Options

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form on a Procedure</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form on a Table or View</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form on a Table with List View</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form on a Table with Report</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This wizard does not support tables having more than 127 columns. Selecting more than 127 columns generates an error.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Creating a Form Using the Create Page Wizard

The easiest way to create a form is to use the Create Page Wizard. As described in Table 11–1, "Create Page Wizard Report Options", this wizard offers numerous form options.

To create a form using the Create Page Wizard:

1. On the Workspace home page, click the Application 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.

See Also: "Managing Master Detail Forms" on page 11-12

See Also: "Managing Tabular Forms" on page 11-6

See Also: "Creating a Form on a Web Service" on page 25-17

See Also: "Creating an Input Form and Report on a Web Service" on page 25-15

See Also: "About Regions" on page 14-4

### Table 11–2 (Cont.) Create Region Wizard Form Options

<table>
<thead>
<tr>
<th>Page Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Detail Form</td>
<td>Builds a two page report and form combination. On the first page, users select a row to update. On the second page, users can update the selected table or view and its associated detail. You choose the tables on which to build the master report and detail form. See Also: &quot;Managing Master Detail Forms&quot; on page 11-12</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>Tabular Form</td>
<td>Creates a form to perform update, insert, and delete operations on multiple rows in a database table. You choose a table on which to build the form. See Also: &quot;Managing Tabular Forms&quot; on page 11-6</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td>Form on a SQL Query</td>
<td>Creates a form based on the columns returned by a SQL query such as an EQUJOIN.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Summary Page</td>
<td>Creates a read-only version of a form. Typically used to provide a confirmation page at the end of a wizard.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form on Web Service</td>
<td>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: &quot;Creating a Form on a Web Service&quot; on page 25-17</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Form and Report on Web Service</td>
<td>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: &quot;Creating an Input Form and Report on a Web Service&quot; on page 25-15</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
</tbody>
</table>
Using a Wizard to Create a Form

[432x756]

5. Select the type of form page you wish to create. See Table 11–1, "Create Page Wizard Report Options".

6. Follow the on-screen instructions.

To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

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 Application 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. See "Using Themes" on page 13-1.
   e. Theme Style - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51. See "Using Theme Styles" on page 13-3.
   f. Click Next.

   Tip: To create an application at this point that consists of just a login page and blank page, click Create Application.

Next, add pages to your application.

5. For Pages, click Add Page.

In the Create Application - Add Page dialog:

a. Select Page Type - Select 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.

e. Repeat the previous steps until all pages have been added.

f. Click Next.
6. Follow the on-screen instructions.

   **Tip:** To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

   **See Also:** "Creating a Database Application" on page 7-4

---

**Creating a Form in Page Designer**

To create a form in Page Designer:

2. Locate the Page Designer Toolbar at the top of the page.
3. On Page Designer Toolbar, click the **Create** menu and select **Create Form Region**.
   The Create Region Wizard appears.
4. Select the type of form you wish to create and click **Next**. To learn more about form options, see "Create Region Wizard Form Options" on page 11-3.
5. Follow the on-screen instructions.

---

**Managing Tabular Forms**

A tabular form enables users to update multiple rows in a table at once from a single page. When created using a wizard, a tabular forms enables you to perform update, insert, and delete operations on multiple rows in a database table. This form includes a built-in multiple row update process that performs optimistic locking behind the scenes to maintain the data integrity.

- About Creating Tabular Forms
- About Running a Tabular Form
- How Tabular Forms Work
- Creating a Tabular Form Using the Create Application Wizard
- Creating a Tabular Form Using the Create Page Wizard
- Synchronizing Tabular Form Column Order with Region Source

---

**About Creating Tabular Forms**

There are two primary methods for creating tabular forms:

- Create Application Wizard
- Create Page Wizard

Tabular forms created using these wizards include automatic DML processes. These wizards create a form that performs update, insert, and delete operations on multiple rows in a database table. Alternatively, you can build a tabular form manually either by creating a updatable report region or standard report region with updatable columns with custom PL/SQL page processes to process the tabular form data.

**Best Practices When Creating Tabular Forms**

Oracle recommends the following best practices when creating tabular forms:
Do not modify the select list of a SQL statement of a tabular form after it has been generated using a wizard. If you do modify the query, make sure the values of the updatable columns are not altered after being queried from the database by the Application Express engine.

Do not add custom JavaScript to display types that use display type specific JavaScript logic (for example, radio groups, simple check boxes, and popup key LOVs).

About Running a Tabular Form

When running a tabular form, data is rendered as a report having updatable columns shown using various form elements (including text fields, text areas, date pickers, select lists, radio groups, and so on).

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.

See Also: "Managing User Interface Defaults" in Oracle Application Express SQL Workshop Guide

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.
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 from, tabular forms do not store data in session state. When a tabular form page is submitted, the tabular form data is processed using built-in data manipulation language (DML), or a custom PL/SQL page processes.

A tabular form with automatic DML has two processes for multi-row operations:
- A Multi Row Update (MRU) process performs create and update operations.
- Multi Row Delete (MRD) process deletes requests on one or more rows.

MRU and MRD processes reference the underlying table name and the primary key columns. The primary key columns must be part of the tabular form SQL query, and the report columns must be either hidden or display-only (save state) in order for the DML processes to be able to identify the correct records. For new records, the primary key values must be `NULL` to be identified as new records. Delete operations are performed by referencing row number of the row to be deleted. Users can check one or more rows on the current page and when they click the Delete button, the row numbers identify the corresponding primary key value(s) and the matching rows are deleted.

Wizard generated tabular forms built-in DML include a checksum column to prevent lost updates. Using a method called optimistic locking, a checksum is computed for each row when rendering a tabular form. This checksum is stored in an extra hidden tabular form column. Upon submission of the form for update, an additional set of checksums is computed on the then current data stored in the database. The new checksums are compared to those built when the tabular form was first rendered, and if there are any mismatches, the update is rejected, because the data in the database was either modified by another user or another process in the time between the tabular form was rendered and the time the data was submitted.

Wizard generated tabular forms also automatically create validations for some columns. Validations are created for columns that are set to `NOT NULL` in the underlying table and columns of type `NUMBER`, `DATE` or `TIMESTAMP`. Note that validations are not created for columns if the column is set to read-only, either based on the user's selection, or defined user interface defaults.

Creating a Tabular Form Using the Create Application Wizard

The Create Application Wizard is designed to quickly create a basic Oracle Application Express application consisting of multiple pages and components. For that reason, this wizard uses general best practices to generate the application and does not include several advanced options included in the Create Page Wizard.

**Tip:** When creating a tabular form, Oracle recommends using a generic column template and avoiding column report templates. See "Report Templates" on page 13-51.

To create a tabular form using the Create Application Wizard:

1. On the Workspace home page, click the **Application 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. See “Using Themes” on page 13-1.
   e. **Theme Style** - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51. See “Using Theme Styles” on page 13-3.
   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. **Select Page Type** - Select **Tabular 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 new page appears at the top of the page.

   e. 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. See “Managing Shared Components” on page 18-1.

7. For Attributes, configure as appropriate and click **Next**.

   **Tip:** To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

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

   **See Also:** "Managing User Interface Defaults" in Oracle Application Express SQL Workshop Guide

**Creating a Tabular Form Using the Create Page Wizard**

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.
Tabular forms perform insert, update and delete operations on table rows. Unlike with the Create Application Wizard, when you create a tabular form using the Create page Wizard you determine how rows are identified. 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. See "Report Templates" on page 13-51.

To create a tabular form using the Create Page Wizard:

1. On the Workspace home page, click the Application 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 Tabular Form.
   The Create Tabular Form Wizard appears.
6. For Table and Columns:
   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 Columns - Select one or more columns to be included in the tabular form. In the pages that follow you define which columns should be updatable.
   d. Allowed Operations - Select the operations to be performed on the table (for example, Update, Insert and Delete).
   e. Use User Interface Defaults - Select whether to use user interface defaults. See field-level Help for more information and "Managing User Interface Defaults" in Oracle Application Express SQL Workshop Guide.
   f. 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.
7. For Primary Key, select one of the following:
   - 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).
8. If you select Managed by Database (ROWID), click Next to continue.
9. 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. Click **Next**.
   d. For Source Type, 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.
      - **Custom PL/SQL function** - Select this option to provide a PL/SQL function to generate returning key value.
      - **Existing sequence** - Select this option to pick the sequence from a list of sequences available in the selected schema.
   e. Click **Next**.

10. For Updatable Columns:
    a. Updatable Columns - Select the updatable columns to include in the form.
    
    You can modify the column order or your SQL query after you create the page.
    b. Click **Next**.

11. For Page and Region Attributes:
    
    **Tip:** To learn more about an attribute, see field-level Help. See “Viewing Field-Level Help” on page 1-21.
    a. Page - Select a page to contain the new updatable report, or enter a new page number.
    b. Page Name - If you are creating a new page, enter a name.
    c. Page Mode - Select a page mode.
    d. Region Title - Enter a name for the region in which the updatable report appears.
    e. Region Template - Select a region template for the updatable report region.
    g. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
    h. Click **Next**.

12. 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.

13. For Buttons and Branching:
    a. Cancel Button Label - Enter text to display on the Cancel button.
    b. Branch to Page - Specify the page number to branch to when the user clicks Cancel.
    c. Apply Changes Button Label - Enter text to display on the Apply Changes button.
Managing Master Detail Forms

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 within a single HTML form. With this form, users can insert, update, and delete values from two tables or views.

When running a master detail form, data is rendered as a report using various form elements (including text fields, text areas, date pickers, select lists, radio groups, and so on). 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.

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 Application Builder icon.
2. Select the application.
3. Select a page containing the tabular form.
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.

Managing Master Detail Forms

This section describes how to create a master detail form.

■ About Master Detail Forms
■ Creating a Master Detail Form Using the Create Application Wizard
■ Creating a Master Detail Form Using the Create Page Wizard

d. Branch to Page - Specify the page number to branch to when the user clicks Submit.
e. Delete Button Label - Enter text to display on the Delete button.
f. Add Row Button Label - Enter text to display on the Add Row button label.
g. Click Next.

14. Click Create.

A success message appears.
On the Master Detail, the master record displays as a standard form and the detail records display in a tabular form at the bottom of the page. In this example, users click **Add Item to Order** to add a new row.

**See Also:** "Managing User Interface Defaults" in *Oracle Application Express SQL Workshop Guide*

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.

**Creating a Master Detail Form Using the Create Application Wizard**

The Create Application Wizard is designed to quickly create a basic Oracle Application Express application consisting of multiple pages and components. This 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 **Application 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. See "Using Themes" on page 13-1.
   - e. **Theme Style** - Select a theme style. Theme styles are only available for newer applications such as theme 42 and 51. See "Using Theme Styles" on page 13-3.
   - 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. Click the **Add Page** button.

   On Add Page:
   - a. **Select Page Type** - Select **Master Detail**.
   - b. **Parent Page** - If applicable, select a subordinate page. To learn more about an attribute, see field-level Help.
   - 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.
Managing Master Detail Forms

Developing Forms 11-15

- Page Mode - Select a page mode.
- Click Add Page.
  The new pages appear at the top of the page.
- 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. See “Managing Shared Components” on page 18-1.

7. For Attributes, configure as appropriate and click Next.

**Tip:** To learn more about an attribute, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

8. Confirm your selections and click Create Application.

Creating a Master Detail Form Using the Create Page Wizard

The Create Page Wizard provides support for creating a master detail 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.

When created using a wizard, the master form on a 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 master detail form using the Create Page Wizard:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. Click Create Page.
4. For Create a Page:
   - User Interface - Select a user interface for the page.
   - Select a page type - Select Form.
   - Click Next.
5. Select Master Detail Form and click Next.
   The Master Detail Wizard appears.
6. On Master Table and Columns:
   - Table/View Owner - Select the schema that owns the master table.
   - Table/View Name - Select the table or view which contains the columns to be included in the master page.
     The columns in that object display.
   - 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 defaults. See field-level Help for more information and "Managing User Interface Defaults" in Oracle Application Express SQL Workshop Guide.

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.

    The columns in that object appear.

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. For Define Primary Key, select one for each table:

- **Managed by Database (ROWID)** - Use the ROWID pseudo column to identify rows to update and delete.

- **Select Primary Key Column(s)** - Use the source table's primary key column(s).

9. If you select **Managed by Database (ROWID)**:

a. Click Next.

b. For Master Options, 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. Click Next.

c. Click Next.

10. If you select **Select Primary Key Column(s)**:

a. On Define Primary Key, select the following for each table:

   - Primary Key Column 1 - Select the first column of the primary key.

   - Primary Key Column 2 - If you have a compound primary key, specify the second primary key column in this attribute. This attribute is optional

   - Click Next.

b. For Primary Key Source, select the method by which the primary key is populated and click Next.

   - **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.
– **Custom PL/SQL function** - Select this option to provide a PL/SQL function to generate returning key value.

– **Existing sequence** - Select this option to pick the sequence from a list of sequences available in the selected schema.

c. For Master Options, specify whether to include master row navigation and click **Next**.

If you include master row navigation, define navigation order columns. If a navigation order column is not defined, the master update form navigates by the primary key column.

d. Click **Next**.

11. On Master Options:

a. Include master row navigation - Master row navigation enables users to navigate the master update form. Select **Yes** to include master row navigation.

b. Master row navigation order - Select the navigation order column. If no column is selected, navigation is ordered by primary key of the master table.

c. Secondary navigation order - Select the secondary navigation order column.

d. Include master report - You can choose to not create a master report page if you have a report page already. Select **Yes** to include a master report page.

e. Click **Next**.

12. On Build Master Detail, select the layout of master detail pages and click **Next**.

Options include

- **Edit detail as tabular form on same page** - Creates a two page master detail.

- **Edit detail on separate page** - Creates a three page master detail.

13. On Page Attributes:

**Tip:** To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

a. Page Attributes - Review and edit the master page and detail page information.

b. Detail Page Detail Region Title - Enter a title for the Detail page.

c. Page Group - Identify the name of the page group you would like to associate with this page.

d. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use. To create breadcrumbs entries:

  – **Parent Entry** - Select a parent entry for this breadcrumb entry.

  – **Entry Name (Master Report)** - Specify the name of the breadcrumb entry for the master report page.

  – **Entry Name (Master Detail Page)** - Specify the name of the breadcrumb entry for the master detail page.

  – **Parent Entry** - This item displays if you have chosen a parent entry.

e. Click **Next**.
14. 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.

15. Click Create.

Creating a Form Manually in Component View

This section describes how to create a form manually.

- About Creating Forms Manually in Component View
- Processing a Form
- Populating Forms in Component View

See Also: "Managing Pages in Component View" on page 9-1

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 Component View" on page 9-5.

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
- Creating a Process that Contains One or More Insert Statements
- Using a PL/SQL API to Process Form Values

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 Component View" on page 9-5
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.

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:

```sql
BEGIN
    DELETE FROM T
    WHERE ID = :P1_ID;
END;
```

**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.

**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 Component View” on page 9-5.
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`.

```sql
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.
Validating User Input in Forms

This section describes how to validate user input in forms.

- About Creating Validations and Error Messages
- Associating a Validation with a Specific Item in Page Designer
- Associating a Validation with a Specific Item in Component View
- About the Error Message Display Location

About Creating Validations and Error Messages

You can use validations to check data a user enters before processing. Once you create a validation and the associated error message, you can associate it with a specific item. You can choose to have validation error messages display inline (that is, on the page where the validation is performed) or on a separate error page.

Creating an inline error message involves these steps:

- Step 1 - Create a validation and specify error message text. To learn more, see "Understanding Validations" on page 17-6.
- Step 2 - Associate the validation with a specific item.

Associating a Validation with a Specific Item in Page Designer

To associate an item with a validation and specify error message text:

2. In Page Designer, select the Processing tree.
3. Under Validating, select the validation item you want to associate.
   The Property Editor changes to Property Editor - Validation.
4. Click the Collapse All icon and then select Show All from the Property Editor Menu.
5. Expand Error and configure the following:
   - Error Message - Enter or edit the 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.
6. Click Save.

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 Component View” on page 9-5.
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.

About the Error Message Display Location

The error message display location identifies where a validation error message displays. Validation error messages can display on an error page or inline within the existing page. Inline error messages can display in a notification area (defined as part of the page template) or within the field label.

To create a hard error that stops processes, including any remaining validations, you must display the error on an error page.

Understanding BLOB Support in Forms

This section describes BLOB support in forms.

- About BLOB Support in Forms
- About Uploading and Downloading Files into a BLOB
- About Displaying the BLOB
- About Removing the Image Reference

About BLOB Support in Forms

If you create a Form (either using the Create Application Wizard, create page of type Form - or Report and Form, or create region of type Form) or add an item to an existing form, any item whose source is a database column of type BLOB results in an item of type File Browse. When the form is called for INSERT, the file selected by the user is loaded into the BLOB column. When the form is called for update, a download link is displayed to the right of the Browse button. Users can use this link to download the file.

About Uploading and Downloading Files into a BLOB

The defaulted BLOB support does not give you all the information a typical application needs to effectively manage a BLOB. In addition to knowing that the column is a BLOB, more information about the file provides a better experience for the end-user. The File Browse page item has additional settings to facilitate managing this additional information completely declaratively.

See Also: File Browse in Appendix A, "About Item Types", item level help for the File Browse settings
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. Do not use it 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:

```sql
ALTER TABLE emp ADD
    (ATTACH_MIMETYPE VARCHAR2(255),
     ATTACH_FILENAME VARCHAR2(255),
     ATTACH_LAST_UPDATE DATE,
     ATTACH_CHARSET VARCHAR2(128));
```

If you manually create a form on a custom table, you can still take advantage of this feature. To do so, use the File Browse item type with a Storage Type setting of BLOB column specified in Item Source Attribute, on a page with a DML Process type of `DML_PROCESS_ROW`. This process determines the table name and primary key columns.

### About Displaying the BLOB

If the BLOB you are working with is an image, you can display it in the form as well. You can use the Display Image item type to handle this declaratively, see Appendix A, "About Item Types" for details. See "Understanding BLOB Support in Forms and Reports" on page 10-89 to handle procedurally.

### 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:

```sql
UPDATE demo_product_info
    SET product_image = NULL,
        MIMETYPE = NULL,
        FILENAME = NULL,
        IMAGE_LAST_UPDATE = NULL,
        CHARSET = NULL
    WHERE product_id = :P6_PRODUCT_ID;
```

---

**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.
This section explains how to create and manage database application components such as calendars, maps, charts, and Help.

- Creating Calendars
- Creating Maps
- Creating Charts
- Creating Help for Your Application

See Also: "Developing Reports" on page 10-1, "Controlling Page Layout" on page 14-1, and "Adding Navigation" on page 16-1

Creating Calendars

This section describes how to create a calendar in Application Builder. Application Builder includes a built-in wizard 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
- Sample Calendar
- Adding a Calendar to a New Page
- Managing Calendar Attributes in Page Designer
- Managing Calendars in Component View
- About the Date Column Format Mask
- Enabling the Dragging and Dropping of Data In an Existing Calendar
- About Supported Calendar Substitution Strings in Legacy Calendars

About Supported Calendars

Application 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 will 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.
- 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.

Sample Calendar
The following is a sample calendar from Sample Database Application.
Creating Calendars

Managing Database Application Components

12-3

This calendar includes the following buttons:

- **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:** Expand the SQL Source region at the bottom of the page to view the underlying SQL.

### Adding a Calendar to a New Page

This section describes how to add a new calendar from the Application home page.

- **Adding a Calendar to a New Page**
- **Adding a Legacy Calendar to a New Page**

**See Also:** "About Supported Calendars" on page 12-1

#### Adding 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 **Application 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. Select Calendar and click Next.

4. For Page and Region Attributes:

   Tip: To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

   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. Region Name - Enter a name for the region to contain the calendar.

   e. Breadcrumb - Select whether to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.

   f. Click Next.

5. 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.

6. 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 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.

      For examples, see field-level Help.

   c. Click Next.

7. For Settings:

   a. Display Column - Select the column which holds the text displayed for events on this calendar.

   b. Start Date Column - Select the column which holds the start date for events displayed on this calendar.

   c. End Date Column - Select the column which holds the end date for events displayed on this calendar. If this attribute is specified, then the calendar displays duration-based events.
Creating Calendars

Managing Database Application Components

12-5

d. Show Time - Select whether the time portion of the date should be displayed.

e. Add Create Page - Identify whether a new form page is created. This form will be used to add events on calendar.

f. Add Edit Page - Identify whether a new form page is created. This form will be used to edit events on calendar.

g. Generate Drag&Drop Code - Specify if the calendar includes code for drag and drop. This code add support for updating the events records when an end user drags and drops events from one date to another.

h. Click Next.

8. If you elected to include a Create/Edit Page, configure the attributes and follow the on-screen instructions.

9. Click Create.

   Tip: If you add a new region of type Calendar directly in Page Designer than you need to specify the SQL Query for the calendar and manually define additional attributes such as Drag and Drop Pl/SQL Code. You will also need to manually define create and edit pages, or link to existing form pages.

Adding a Legacy Calendar to a New Page

In previous releases Legacy Calendar was referred to as Calendar (Template), Classic calendar, Easy calendar, and SQL calendar). A Legacy Calendars are template-based and will be deprecated in the future release.

   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. See “Enabling the Dragging and Dropping of Data In an Existing Calendar” on page 12-12.

To create a Legacy Calendar by running the Create Page Wizard:

   Tip: To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

1. Run the Create Page Wizard:

   a. Navigate to the Workspace home page.

   b. Click the Application 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. Select Calendar (Template) and click Next.

4. For Page Attributes:
Creating Calendars

1. 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.

2. Page Name - Enter a title for the page to be created.

3. Page Mode - Select a page mode.

4. Region Template - Select a region template for the new calendar region. The template controls the appearance of the region.

5. Region Name - Enter a name for the region to contain the calendar.

6. 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.

7. Breadcrumb - Select whether to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.

8. Click Next.

5. 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.

6. For Source, select if the calendar source is a table or SQL query.

   If the Calendar source is table:
   a. Source - Select Table.
   b. Owner - Select the database schema that owns the table from which you are creating a calendar.
   c. Table/View Name - Specify the table to be used in the calendar.
   d. Click Next.

   If the Calendar source is SQL Query:
   a. Source - Select SQL Query.
   b. 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.
   c. Click Next.

7. 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.
Creating Calendars

Managing Database Application Components

– 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.

8. For Link Details:

a. 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.

To learn more, see field-level Help.

b. Click Next.

9. Click Create.

Managing Calendar Attributes in Page Designer

Tip: When creating a calendar based on a SQL query, you must manually create an On Demand page process to enable drag and drop functionality. See "Enabling the Dragging and Dropping of Data In an Existing Calendar" on page 12-12.

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
- Editing an Existing Calendar to Include Add and Edit Functionality

See Also: "Managing Calendars in Component View" on page 12-9

Editing Calendar Attributes in the Property Editor

To edit calendar attributes:


Page Designer appears

2. In the Rendering tree, select the calendar title.

The Property Editor displays the region attributes.

3. In the Rendering tree, under the calendar title, select the Attributes node.

The Property Editor displays the Calendar attributes.

4. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

5. Edit the region attributes:
6. Click **Save**.

**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:


Page Designer appears

2. In the Rendering tree, under the calendar title, select the Attributes node.

The Property Editor displays the Calendar attributes.

3. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

   **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 Create Link or View / Edit Link:
   a. Open Link In - Specify if the page or URL replaces the current page or opens in a dialog.
   b. Create Target - Click **No Link Defined**.

5. In the Link Builder - Event Target dialog:
   a. Type - Select the type of target application: **Page in this application** or **Page in a different application**.
   b. Application - If you select **Page in a different application**, select the application.
   c. Page - Specify the target page number.
   d. Set Items:
      - Name - For Create Link enter start date column. For duration-based events also enter the end date column. For View/Edit Link enter the primary key column.

   **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.
Creating Calendars

Managing Calendar Attributes in Component View

This section describes how to manage calendars in Component View.

- Creating a Calendar in Component View
- Editing Calendar Attributes in Component View

Tip: When creating a calendar based on a SQL query, you must manually create an On Demand page process to enable drag and drop functionality. See “Enabling the Dragging and Dropping of Data In an Existing Calendar” on page 12-12.

See Also: "Managing Calendar Attributes in Page Designer" on page 12-7

Creating a Calendar in Component View

To create a Calendar in Component View:

1. Create a Calendar region:
   b. Create a new Calendar region. Under Region, click the Create icon.
      The Create Region Wizard appears.
   c. Select Calendar and click Next.

2. For Implementation, select Calendar and click Next.

3. 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.

4. 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 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. For examples, see field-level Help.

c. Click **Next**.

5. For Settings, accept the defaults and click **Next**.

To learn more about an attribute, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

6. 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**.

7. Click **Create**.

**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.

- Accessing Calendar Attributes
- Editing an Existing Calendar to Include Add and Edit Functionality

**Accessing Calendar Attributes** To access Calendar attributes:

1. Navigate to the appropriate page. See “Viewing a Page in Component View” on page 9-5.

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. See "Viewing Field-Level Help" on page 1-21.

5. Click Apply Changes.

**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. See "Accessing Calendar Attributes" on page 12-10.

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.

**About the Date Column Format Mask**

Because the calendar passes the date in specific format, the selected date column must have a format mask of RRRRMMDDHH24MISS. 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. The default value of the date column should be changed to:
Consider the following example:

```
return to_char(to_date(:P130_HIREDATE_TMP,'RRRRMMDDHH24MISS'),:APP_DATE_TIME_)

The source type for default should be PL/SQL Function returning value.

Enabling the Dragging and Dropping of Data In an Existing Calendar

If you choose to enable drag and drop capability when you create a Desktop calendar, the generation process automatically defines PL/SQL code to implement this functionality.

For Legacy Calendars an On Demand process is required to enable the drag and drop functionality. This process is automatically created if you use the Create Page Wizard. In contrast, when you create a Legacy Calendar on a new page by running the Create Page Wizard, the Enable Drag and Drop option does not appear. To enable this functionality, you must create the On Demand process, create a 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 Legacy Calendar data:

1. Edit the attributes for the calendar.
2. Set Drag and Drop to Yes.
3. Enter the appropriate PL/SQL Code using the substitution variables APEX$NEW_ START_DATE, APEX$NEW_END_DATE and APEX$PK_VALUE.

For example, for a duration based calendar use:

```
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;
```

About Supported Calendar Substitution Strings in Legacy Calendars

Application 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 Strings list on the right side of the page.

See Also: "Using Themes and Theme Styles" on page 13-1

Creating Maps

Application Builder includes built-in wizards for generating map charts on pages with a Desktop user interface. How you create a map chart depends upon whether you are adding the map to an existing page, or adding a map on a new page.
About Map Chart Support

Map chart support in Oracle Application Express is based on the AnyChart AnyMap Interactive Maps Component. 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](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*:


About Creating SQL Queries for Maps

You define a map in Application Builder using a wizard. You select a map type, map source, and provide a SQL query using the following syntax:

```sql
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 Application Builder icon.
2. Select an application.
3. Click Create Page.
4. Select Map Chart and click Next.
5. For Map Type, select United States of America and click Next.
6. For Map Source, expand Country Maps and select States.
7. For Page Attributes specify the appropriate attributes and 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 Map Attributes, select the appropriate attributes and click Next.

The Query page appears.

10. Scroll down and expand and review the May Query Example and Map Reference Information regions.

11. Scroll down and expand and review the Map Reference Information region at the bottom of the page.
12. Specify a SQL query by either:
   - Entering a SQL query in the field provided. See "About Creating SQL Queries for Maps" on page 12-23.
   - 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:

```sql
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/

13. Click Next.
14. Click Finish.
Adding Map Charts to a New Page

To add a map chart on a new page:

1. On the Workspace home page, click the Application 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 Map Chart.
   c. Click Next.
5. Select a map type (for example, Europe), and click Next.
6. 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).
7. 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. Region Template - Select a region template for the region in which this map is to appear.
   e. Region Name - Enter a name for the region.
   f. Navigation List - New pages can be added to an existing navigation list, either as top level entries, or as links in the navigation list hierarchy. Click Configure to add your page to the navigation list.
   g. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
      To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
   h. 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 Map Attributes, select the appropriate attributes and click Next.
   To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
10. Specify a query by either:
    - Entering a SQL query in the field provided. See "About Creating SQL Queries for Maps" on page 12-13.
    - 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 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. See “About Creating SQL Queries for Maps” on page 12-13.

11. Click Next.
12. Click Create.

Managing Map Charts in Page Designer

This section describes how to manage map charts in Page Designer.

- Adding a Map Chart in Page Designer
- Editing Map Chart Attributes in Page Designer
- Using Custom XML with Map Charts
- Enabling Automatic Updates

Adding a Map Chart in Page Designer

To add a map chart to an existing page.


Page Designer appears

2. From the Gallery select the Map Chart region and drag it to the appropriate location in Grid Layout.

**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 Property Editor - Region, edit the region attributes. Required attributes display a red triangle in the upper left corner next above attribute label.

**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 tree, under the map chart title, select the Attributes node.

5. In the Property Editor, edit the following Map Chart 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.

6. Click Save.

**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:

   Page Designer appears.

2. To edit region attributes:
   a. In the Rendering tree, select the region that contain 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 tree, 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" on page 12-19 and "Enabling Automatic Updates" on page 12-19

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:
   Page Designer appears.
2. In the Rendering tree, under the region title, select the Attributes node.
   The map chart attributes display in the Property Editor.
3. Expand the Custom XML section.
4. For Custom XML, Custom, select Yes.
5. Edit the XML.
6. Click Save.

   Tip: For more information on supported XML format for maps, see the Online Map XML Reference at:
   http://6.anychart.com/products/anymap/docs/

Enabling Automatic Updates
You can create map charts that monitor information by enabling the Asynchronous Update attribute on the Map Attributes page. Enabling this attribute updates the map to reflect changes in the underlying data within a specified time interval.

To enable automatic refresh:
   Page Designer appears.
2. In the Rendering tree, under the region title, select the Attributes node.
   The map chart attributes display in the Property Editor.
3. Expand the Automatic Refresh section.
   a. Automatic Refresh - Select Yes.
   b. Interval - Enter the interval in seconds between chart updates.
      Oracle discourages very small updates intervals, such as 2 seconds, since they may cause serious database performance issues.
4. Click Save.

Managing Map Charts in Component View
This section describes how to manage map charts in the Component View.

- Adding a Map Chart in Component View
Creating Maps

- Editing Map Chart Attributes
- Using Custom XML with Map Charts
- Enabling Asynchronous Updates

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 Component View" on page 9-5.
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" on page 1-21.
8. Specify a query by either:
   - Entering a SQL query in the field provided. See "About Creating SQL Queries for Maps" on page 12-13.
   - 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" on page 12-13.
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.

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 Component View" on page 9-5.
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" on page 1-21.
4. Click Apply Changes.

See Also: "Using Custom XML with Map Charts" on page 12-21 and "Enabling Asynchronous Updates" on page 12-22

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.
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/
**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.
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.

---

**Creating Charts**

This section describes how to create charts in Oracle Application Express. Application Builder includes built-in wizards for generating Flash and HTML5 charts, and supports the editing of SVG and HTML charts. Oracle Application Express supports only these four types of graphical charts.

- About Chart Types
- Creating SQL Queries for Charts
- Adding a Chart to a New Page
- Managing Charts in Page Designer
- Managing Charts in Component View
- Understanding SVG Chart Support
- Understanding HTML Chart Support
- About Saving Flash Charts
- Using Custom XML with Flash and HTML5 Charts

**About Chart Types**

This section describes the different types of charts you can create in Application Builder.

- About HTML5 Charts
- About Flash Charts

**About HTML5 Charts**

HTML5 chart support in Oracle Application Express is based on the AnyChart HTML5 Chart Component. AnyChart is a flexible Flash and JavaScript (HTML5) based solution that enables developers to create animated and compact interactive charts. 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

To learn more see:
http://6.anychart.com/products/anychart/docs/

About Flash Charts
Flash chart support in Oracle Application Express is based on the AnyChart Flash
Chart Component. AnyChart is a flexible Flash and JavaScript-based solution that
enables developers to create animated, compact, interactive charts. Flash charts are
rendered by a browser and require Flash player 9 or later. With AnyChart 6.0, Flash
charts are rendered using the FLASH_PREFERRED rendering type. If Flash Player is
not available on the device displaying the chart (for example, pages displayed on an
iPhone), AnyChart automatically switches to the HTML5 chart engine and displays an
SVG-based chart.

Creating SQL Queries for Charts
This section describes the supported syntax for creating charts in Application Builder.

- Standard Chart Syntax
- Dial Chart Syntax (Flash and HTML5)
- Multiple Series Syntax (Flash and HTML5)
- Range Chart Syntax (Flash and HTML5)
- Scatter Chart Syntax (Flash and HTML5)
- Candlestick Chart Syntax (Flash and HTML5)
- Gantt Chart Syntax (Flash Only)

Standard Chart Syntax
You define a chart in Application Builder using a wizard. For most chart wizards, you
select a chart type and provide a SQL query using the following syntax:

```sql
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:

```sql
SELECT null, last_name, salary
FROM   employees
WHERE  DEPARTMENT_ID = :P101_DEPARTMENT_ID
```
Creating Charts

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
```

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   ...
```

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   ...
```

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   ...
```

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   ...
```

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
Creating Charts

start date and planned end date can also be used. To create a Project Gantt chart, create a Flash chart and provide a SQL query using the following syntax:

```
SELECT link, task_name, task_id, parent_task_id, actual_start_date, actual_end_date, progress
FROM ...
```

Resource Gantt charts require a resource id, resource name, parent resource id, actual start date, and actual end date value for each task. To create a Resource Gantt chart, create a Flash chart and provide a SQL query using the following syntax:

```
SELECT link, resource_id, resource_name, resource_parent_id, actual_start_date, actual_end_date
FROM ...
```

To represent parent-child hierarchical data on a Resource Gantt chart, provide a SQL query using START WITH.CONNECT BY syntax:

```
SELECT link, resource_id, resource_name, resource_parent_id, actual_start_date, actual_end_date
FROM ...
START WITH resource_parent_id IS NULL
CONNECT BY PRIOR resource_id = resource_parent_id
ORDER SIBLINGS BY resource_name
```

Adding a Chart to a New Page

To add a chart on a new page:

1. On the Workspace home page, click the Application 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 Rendering - Select a chart type. See "About Chart Types" on page 12-22.
   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 Page and Region 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.
To learn more, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

d. Page Group - Identify the name of the page group you would like to associate with this page.

e. Region Template - Select a region template for the region in which this chart is to appear.

f. Region Name - Enter a name for the region.

g. Navigation List - New pages can be added to an existing navigation list, either as top level entries, or as links in the navigation list hierarchy. Click **Configure** to add your page to the navigation list.

h. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page.

i. 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. For Chart Attributes, specify the appropriate attributes and click **Next**.

To learn more about an attribute, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

9. For Query:

   a. Specify a query by either:
      - Entering a SQL query in the field provided. See “Creating SQL Queries for Charts” on page 12-23.
      - 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.

   c. Click **Next**.

10. Click **Create**.

### Managing Charts in Page Designer

This section describes how to create and edit charts in Page Designer.

- Adding a Chart in Page Designer
- Editing Chart Attributes in Page Designer
- Switching an HTML5 Chart to a Flash Chart
- Enabling Asynchronous Updates

### 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. See “Viewing a Page in Page Designer” on page 8-15.
2. From the Gallery select the Chart region and drag it to the appropriate location in Grid Layout.

**Tip:** You also right-click Tree 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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

4. In Property Editor - Region, edit the required attributes.

Required attributes display a red triangle in the upper left corner next above attribute label.

**Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Edit the chart attributes:

   a. In the Rendering tree, select the **Attributes** node.

   Property Editor - Attributes appears.

   b. Edit the required chart attributes.

   Required attributes display a red triangle in the upper left corner next above attribute label.

   c. Click **Next**.

6. Edit the series:

   a. In the Rendering tree, select the **Series** node.

   Property Editor - Series appears.

   b. Edit the required chart attributes.

   c. Click **Next**.

7. Click **Save**.

**Editing Chart Attributes in Page Designer**

Once you have created a chart, you can alter its display by editing chart attributes on the Chart Attributes page.

**Note:** The deprecated chart type SVG cannot be edited using Page Designer.
Creating Charts

To access the Chart Attributes page:

1. View the page to contain the chart in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.

2. In the Rendering tree, select the Attributes node.
   Property Editor - Attributes appears.

3. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

4. Edit the chart attributes.
   Required attributes display a red triangle in the upper left corner next above attribute label.

5. Click Save.

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. See "Viewing a Page in Page Designer" on page 8-15.

2. In the Rendering tree, 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. See “Viewing Field-Level Help” on page 1-21.

5. Click Save.

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
- Creating a Refresh Dynamic Action

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. See "Viewing a Page in Page Designer" on page 8-15.

2. In the Rendering tree, select the Attributes node.
   Property Editor - Attributes appears.

3. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.
4. For Automatic Refresh, select Yes.
5. Click Save.

**Creating a Refresh Dynamic Action in Page Designer** To create a Refresh dynamic action:
1. View the page to contain the item in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.
2. Click the Dynamic Actions accordion on the left side of the page, beneath the Rendering tree.
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. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.
5. Under Property Editor - Dynamic Actions, 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. 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.
   e. Click Next.

   Next, define the action that to be performed if the event evaluates to True.
6. In the Rendering tree, select the first action under True:
   a. Action - Select Refresh.
   b. Selection Type - Select Region.
   c. Region - Select the region containing the chart.
7. Click Save.

**Managing Charts in Component View**

This section describes how to create and manage charts in Component View.

- Adding an Chart in Component View
- Switching an HTML5 Chart to a Flash Chart
- Editing Chart Attributes in Component View
- Enabling Asynchronous Updates
Adding an Chart in Component View

To add a chart to an existing page in Component View:


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. See “About Chart Types” on page 12-22.
   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. See "Viewing Field-Level Help" on page 1-21.
   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. See “Creating SQL Queries for Charts” on page 12-23.
      - 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. See "Understanding Conditional Rendering and Processing” on page 2-12.
   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” on page 20-49.
   c. Click Create Region.
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. See "Viewing a Page in Component View" on page 9-5.
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. See "Viewing Field-Level Help" on page 1-21.
4. Click Apply Changes.

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. See "Viewing a Page in Component View" on page 9-5.
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. See "Viewing Field-Level Help" on page 1-21.
5. Click Apply Changes.

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
- Creating a Refresh Dynamic Action

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. See "Editing Chart Attributes in Component View" on page 12-31.
   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.

Creating a Refresh Dynamic Action To create a Refresh dynamic action:
1. Navigate to the page containing the chart. See "Viewing a Page in Component View" on page 9-5.
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.

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

This section describes restrictions that apply when upgrading SVG charts to HTML5 and also describes the instructions for upgrading.

- About SVG Chart Upgrade Restrictions
- Upgrading an SVG Chart to HTML5
- Upgrading all SVG Charts in an Application to HTML5

You can automatically upgrade single or multiple SVG charts to HTML5.

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.

Upgrading an SVG Chart to HTML5  To upgrade an SVG chart to HTML5:

1. Navigate to the page. See "Viewing a Page in Component View" on page 9-5.
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.

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 Application Builder icon.
   The Application 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.
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.

```xml
<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;opacity:.5;}
#dial.alert{fill:#FF0000;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 12–1 describes all supported CSS classes. Note that certain classes only apply to specific chart types.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>Defines the appearance of text that displays in a chart.</td>
</tr>
<tr>
<td>tspan</td>
<td>Defines the appearance of text that displays in a chart. tspan should match the definition of text.</td>
</tr>
<tr>
<td>text.title</td>
<td>Overrides the default chart text. Use this class for title text.</td>
</tr>
<tr>
<td>text.moredatafound</td>
<td>Defines the appearance of more datafound text.</td>
</tr>
<tr>
<td>rect.legend</td>
<td>Creates the rectangular box that holds the chart legend.</td>
</tr>
</tbody>
</table>

To remove the legend border, change `rect.legend` to the following:
```xml
rect.legend{fill:#CCCC99;stroke:none;}
```
Table 12–1  (Cont.) Available SVG Chart CSS Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text.legend</td>
<td>Defines the text that appears in the chart legend.</td>
</tr>
<tr>
<td>#background</td>
<td>Creates the entire background for the SVG plug-in.</td>
</tr>
<tr>
<td></td>
<td>For a solid white background with no border, change #background to the following:</td>
</tr>
<tr>
<td></td>
<td>#background{fill:#FFFFFF;stroke:#FFFFFF;stroke-width:2;}</td>
</tr>
<tr>
<td>rect.chartholderbackground</td>
<td>Not applicable to pie and dial charts. Creates the background of the rectangle that holds the chart data.</td>
</tr>
<tr>
<td></td>
<td>For a clear background, change rect.chartholderbackground to the following:</td>
</tr>
<tr>
<td></td>
<td>rect.chartholderbackground(display:none;);</td>
</tr>
<tr>
<td>#timestamp</td>
<td>Only applicable if the Asynchronous Update chart attribute is set to Yes. Controls the appearance of the update timestamp test.</td>
</tr>
<tr>
<td></td>
<td>To disable the display of the timestamp, use defines #timestamp as follows in the Custom CSS, Inline attribute.</td>
</tr>
<tr>
<td></td>
<td>&quot;#timestamp{display:none;}&quot;</td>
</tr>
<tr>
<td>text.tic</td>
<td>Dial charts only. Defines the numbers on a dial chart.</td>
</tr>
<tr>
<td>line.tic</td>
<td>Dial charts only. Defines the graduation mark that displays directly beneath the number on a dial chart.</td>
</tr>
<tr>
<td>#dial</td>
<td>Dial charts only. Defines the value that displays on the dial chart.</td>
</tr>
<tr>
<td>#dial.alert</td>
<td>Dial charts only. Defines a value (called an alert value) that renders in a dial chart using a different display.</td>
</tr>
<tr>
<td>#dialbackground</td>
<td>Dial charts only. Creates the background of a dial chart.</td>
</tr>
<tr>
<td>#dialcenter</td>
<td>Dial charts only. Creates the center of the dial on a dial chart.</td>
</tr>
<tr>
<td>#dialbackground-border</td>
<td>Dial charts only. Works with #dialbackground to create specific graphic effect.</td>
</tr>
<tr>
<td>#low</td>
<td>Dial charts only. Defines the historical low watermark of the data being displayed on a chart.</td>
</tr>
<tr>
<td>#high</td>
<td>Dial charts only. Defines the historical high watermark of the data being displayed on a chart.</td>
</tr>
<tr>
<td>#XAxisTitle</td>
<td>Defines the title that appears on the x-axis</td>
</tr>
<tr>
<td>#YAxisTitle</td>
<td>Defines the title that appears on the y-axis</td>
</tr>
<tr>
<td>.XAxisValue</td>
<td>Defines the value that appears on the x-axis</td>
</tr>
<tr>
<td>.YAxisValue</td>
<td>Defines the value that appears on the y-axis</td>
</tr>
<tr>
<td>.AxisLabel</td>
<td>Similar to the axis value.</td>
</tr>
<tr>
<td>.nodatafound</td>
<td>Defines the text element that displays if no information is available.</td>
</tr>
<tr>
<td>.AxisLine</td>
<td>Indicates zero on charts that have negative values.</td>
</tr>
</tbody>
</table>
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:


2. Navigate to the page containing the chart. See "Viewing a Page in Component View" on page 9-5.

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.

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. See "Understanding SVG Chart Support" on page 12-32.


3. Access the Chart Attributes page:

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.GridLine</td>
<td>Creates the horizontal and vertical lines on the chart.</td>
</tr>
<tr>
<td>g.dataholder rect</td>
<td>Applies a blanket style to all data that displays in the chart.</td>
</tr>
<tr>
<td>.legenditem rect</td>
<td>Applies a blanket style to all rectangular items in the legend.</td>
</tr>
</tbody>
</table>

Table 12–1 (Cont.) Available SVG Chart CSS Classes
Creating Charts

Managing Database Application Components

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.

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. See "Viewing a Page in Component View" on page 9-5.

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;}

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

- Upgrading All HTML Charts to HTML5

See Also: "Region Type Plug-ins, HTML Chart at:


Upgrading an HTML Chart to HTML5

To upgrade an HTML chart to HTML5:

1. On the Workspace home page, click the Application Builder icon.
   
The Application Builder home page appears.

2. Select an application.
   
The Application home page appears.

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.

Upgrading All HTML Charts to HTML5
To upgrade all HTML charts to HTML5:
1. On the Workspace home page, click the Application Builder icon.
   The Application 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.

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.

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/

Creating Help for Your Application

Help created in Application 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
- Creating Help in Component View
- Editing Multiple Item Help Topics Simultaneously
- Creating a Help Navigation Bar Entry

Creating Help in Page Designer

This section describes how to create Help in Page Designer.

- About Creating Help
- Creating a Help Page and Region in Page Designer
- Defining Page Help in Page Designer
- Defining Item Help in Page Designer

About Creating Help

Application Builder includes built-in attributes to create Help for your application. Creating Help for your application involves the following steps:

- Step 4: Create a navigation bar icon to link to your Help page. See “Creating a Help Navigation Bar Entry” on page 12-44.

Creating a Help Page and Region in Page Designer

To create a Help page and region in Page Designer:


   Page Designer appears.

3. From the Gallery select the Help Text region and drag it to the appropriate location in Grid Layout.

   Tip: You also right-click Help Text region, select Add To, and select the appropriate location.

4. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

5. Edit the region 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 title for the region.
   b. Layout - Specify the Sequence, Parent Region, and Position.
   c. Appearance - Specify the Template and Item Display Position.

6. Click Save.

Defining Page Help in Page Designer

To define Page Help text in Page Designer:


   Page Designer appears.

2. In the Rendering tree, click the page name.

3. In Property Editor - Page, scroll down to Help and enter text in Help Text attribute.

   Tip: By default, Help is populated with the text No help is available for this page.

4. Click Save.

Defining Item Help in Page Designer

To define Page Help text in Page Designer:


   Page Designer appears.

2. In the Rendering tree, click item for which you want to define Help.

3. In Property Editor - Page Item, scroll down to Help and enter text in Help Text attribute.

4. Click Save.
Creating Help for Your Application

Managing Database Application Components

Repeat the previous procedure for each item requiring Help text.

Creating Help in Component View

This section describes how to create Help in Component View.

- About Creating Help
- Creating a Help Page and Region in Component View
- Defining Page Help in Component View
- Defining Item Help in Component View

About Creating Help

Application 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" on page 12-41.
- **Step 2:** Define page Help text. See "Defining Page Help in Component View" on page 12-42.
- **Step 3:** Define item Help text. See "Defining Item Help in Component View" on page 12-42.
- **Step 4:** Create a navigation bar icon to link to your Help page. See "Creating a Help Navigation Bar Entry" on page 12-44.

Help created in Application 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 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:

\[ f?p=&APP_ID.:\text{Page}:\text{Session}:&APP_PAGE_ID. \]

To create a Help page and region in Component View:

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" on page 2-12.
   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" on page 20-49.
   c. Click Next.

9. Click Create Region.

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 Component View" on page 9-5.
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.

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 Component View" on page 9-5.
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.
Editing Multiple Item Help Topics Simultaneously

This section explains various ways to edit multiple Help topics at once.

- Accessing the Bulk Edit Item Help Report
- Seeding Item Help Topics
- Searching for Existing Item Help Topics
- Searching for an Item Label

Accessing the Bulk Edit Item Help Report

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 Application 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.

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. Access the Bulk Edit Item Help report as described in "Accessing the Bulk Edit Item Help Report" on page 12-43.
2. Click Seed Item Help Text.
3. In Default Help Text, enter the default text to appear in all Help topics.
4. Click Apply Changes.

Searching for Existing Item Help Topics

You can search for existing Help text, or for an item label.

To search for existing Help topic:

1. Access the Bulk Edit Item Help report as described in "Accessing the Bulk Edit Item Help Report" on page 12-43.
2. In Help Contains, enter keywords.
3. Click Go.
Searching for an Item Label

To search for an item label:

1. Access the Bulk Edit Item Help report as described in "Accessing the Bulk Edit Item Help Report" on page 12-43.
2. In Help Contains, enter keywords.
3. Click Go.

See Also: "Accessing Application Utilities" on page 7-61

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 Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
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" on page 1-21.
   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.
   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" on page 16-15
This section describes different ways you can alter your database application’s user interface and page layout through themes and templates.

- Using Themes
- Managing Themes and Subscriptions
- Creating Custom Themes
- Using Custom Cascading Style Sheets

Using Themes

This section explains how to use themes and theme styles when building an Oracle Application Express application.

- About Themes
- About Responsive Design
- About Grid-based Themes
- Using Theme Styles
- Understanding Template Options
- Accessing the Themes Page
- Editing Themes

See Also: "Managing Themes and Subscriptions" on page 13-13 and "Managing Themes and Subscriptions" on page 13-13

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.
Using Themes

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" on page 13-19.

- **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" on page 13-3

---

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" on page 3-1 to view applications using *Universal Theme* - 42

---

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 such as *Theme 26 - Productivity Applications*, use a table-based grid. In table-based themes, regions can be positioned next to each one another and forms can have two or more items in a single row. Newer themes, such a *Universal Theme* - 42, use a DIV-based grid. Pages using a DIV-based grid offer the following advantages:
Using Themes and Theme Styles

■ 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" on page 13-16

**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**

■ **Creating a Theme Style**

■ **Editing a Theme Style**

■ **Using Theme Roller**

**About Theme Styles**

A theme style defines a CSS style sheet that is added to the base CSS to alter the look and feel of an application. Newer themes such as *Universal Theme - 42* 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, you can select theme style by:

■ Editing the Theme Styles attribute when running the Create Application Wizard. See "Using the Create Application Wizard" on page 7-1.

■ Editing the User Interface Details, Attributes, Theme Styles. See "Managing the Application User Interface" on page 7-29.

■ Accessing the Create/Edit Theme page and editing, Styles. See "Editing a Theme" on page 13-8 and "Styles" on page 13-12.

**Creating a Theme Style**

To create a theme style:


2. Select a theme and click the **Styles** tab.

3. Click **Add Style**.

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.
Using Themes

5. For Theme Roller Attributes:
   a. Read Only - Select Yes to prevent any updates to the theme style. Select No to allow 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. JSON Configuration - The JSON configuration is generated by Theme Roller when the theme style is saved.
      Oracle does not recommend manually updating the configuration. However, you can copy a JSON configuration from another theme style, such as a theme style in another application, to manually update the configuration.

6. Click Create.

Editing a Theme Style
To edit a Theme Style:
2. Select a theme and click the Styles tab.
3. Select a Theme Style.
   The Theme Styles page appears.
4. 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.
5. To delete the current theme style, click Delete.
6. Click Create.

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: Run a packaged application that uses the Universal Theme - 42 to see an example of Theme Roller. See "Utilizing Packaged Applications" on page 3-1.

To use Theme Roller:
1. Preview the page by running it. See "Running an Application or Page" on page 2-9.
   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" on page 13-8 and "Styles" on page 13-12.
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.
Using Themes

- **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. To select a new style, choose an existing style from the Style list.

5. To change all the colors of the theme style at once, drag the circles in the color wheel. You can choose between making your theme style dependent on one color (monochromatic) or two (dual).

6. To edit a specific component, expand a group and click on the color or styling of the component you wish to edit.

   For attributes that support the selection of foreground and background colors, the check box preceding the two color selections indicates color contrast information based on Web Content Accessibility Guidelines (WCAG) 2.0. A higher calculated contrast score indicates that the color combination is more accessible.

   **Tip:** to set the theme style you are working as the current one for the application, click **Set as Current**.

7. To set the theme style you are working as the current one for the application.

8. Click **Save** to commit your changes to the server. If the selected theme is read only, then click **Save As** to save your changes as a new theme.

Understanding Template Options

Template options provide a declarative way for developers to apply different styles to components on an Oracle Application Express page.

- **About Template Options**
- **About Default Template Options**

See Also: "Using Template Options" on page 13-26

**About Template Options**

Template options enable developers to declaratively apply CSS modifiers to the templates they have chosen for pages, regions, reports, lists, breadcrumbs, items and buttons. A CSS modifier is a reference to a CSS class defined in a CSS style sheet. CSS modifiers enable a developer to use the same HTML markup but present it in a variety of different styles simply by applying a CSS. Template options reduce the need to have a large number of nearly identical templates defined in a theme in order to achieve different styles for button colors, region widths, font settings, item label markup, and so on.

Examples of how developers can use template options include:

- Applying different colors or accents
- Applying different spacing and padding
- Rendering buttons in different styles, with and without icons
- Displaying form fields with different alignments.

Without template options, applying these variations would require a large number of nearly identical templates, or in-depth knowledge of CSS. Components that support...
template options include: pages; regions; classic reports; breadcrumbs; lists; items and labels; and buttons.

**About Default Template Options**

Developers can select template options that are not part of a template option group to be the default for a template. Application Builder automatically applies default template options by rendering the component that references the template. Default template options are not written to an actual component. Instead, a #DEFAULT# substitution string is defined for the component, thus enabling developers to centrally modify the template option defaults.

Developers can choose not to use the template option defaults, which removes the #DEFAULT# substitution string from the component. If template option defaults are disabled, then developers can selectively apply available template options to a component. All template options that are selected directly when editing the component are written to the component’s metadata, represented by their corresponding CSS classes string. At runtime, the selected template options and default template option are part of the default are combined with the component’s CSS classes string and applied to the component.

**About Template Option Groups and Presets**

Developers can define the purpose of related template options by creating template option groups. Examples of template option groups include Button Size, Button Style, List Examples, List Style, and Form Label Position. When developers create a template option group, they can specify a template option preset at the template-level. For example, suppose you have a button template that has the template options Large and Small. If you define Small as the preset, then any new button referencing the template would automatically have the template option Small applied to it when is it created.

Template options that are part of a group are optional unless a present is defined at the template-level.

**Accessing the Themes Page**

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” on page 13-8.

To access the Themes page:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. Click Shared Components.
4. Under User Interface, select Themes.
   
   The Themes page appears.
5. Click the View Report icon.
   
   The Themes page appears as a report. A check mark in the Is Current column indicates which theme is selected.
6. Click the theme name.
   
   The Edit Theme page appears.
Editing Themes

This section describes how to edit a theme attributes.

- Editing a Theme
- Theme Attributes

See Also: "Creating Custom Themes" on page 13-23

Editing a Theme

To review or edit a theme:

2. Click the theme name.
   The Create/Edit Theme page appears and is divided into sections.
3. Edit the appropriate attributes.
   To learn more, see "Theme Attributes" on page 13-8.
4. To save any changes, click Apply Changes.

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
- Theme Subscription
- JavaScript and Cascading Style Sheets
- Component Defaults
- Region Defaults
- Dialog Defaults
- Transition Defaults
- Global Templates Options
- Icons
- Image
- Styles
- Files

Name  Table 13–1 describes the attributes under Name.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Identifies the current application ID. See Also: &quot;Editing the Application Attributes&quot; on page 7-14</td>
</tr>
<tr>
<td>Theme Number</td>
<td>Identifies the number of a theme. A theme is a collection of templates which define the user interface of an application.</td>
</tr>
<tr>
<td>Name</td>
<td>Enter a short descriptive name for the theme.</td>
</tr>
</tbody>
</table>
### Using Themes and Theme Styles

#### 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.

#### JavaScript and Cascading Style Sheets
Table 13–2 describes the attributes under JavaScript and Cascading Style Sheets.

#### Table 13–2  JavaScript and Cascading Style Sheets

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript File URLs</td>
<td>Enter JavaScript file URLs for code to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file, you can use the substitution string #MIN# to include .min or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. JavaScript file URLs you enter here replaces the #THEME_JAVASCRIPT# substitution string in the page template. To view examples, see field-level Help.</td>
</tr>
<tr>
<td>CSS File URLs</td>
<td>Enter Cascading Style Sheet file URLs to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string #MIN# to include .min or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string #APP_VERSION# if you want to include the application's version in the file URL. File URLs you enter here will replace the #THEME_CSS# substitution string in the page template. To view examples, see field-level Help.</td>
</tr>
</tbody>
</table>

#### Component Defaults
Table 13–3 describes the default templates by component type.

#### Table 13–3  Component Default Templates

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>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” on page 9-12</td>
</tr>
</tbody>
</table>
Using Themes

13-10  Oracle Application Express Application Builder User's Guide

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation Bar List</td>
<td>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.</td>
</tr>
<tr>
<td>See Also: &quot;Display Attributes&quot; on page 9-12</td>
<td></td>
</tr>
<tr>
<td>Navigation Menu List Position</td>
<td>Specifies the default navigation menu list position.</td>
</tr>
<tr>
<td>Navigation Menu List (Top)</td>
<td>Specifies the default navigation menu list template used when you create a new list to be displayed on the top of the page.</td>
</tr>
<tr>
<td>Navigation Menu List (Side)</td>
<td>Specifies the default navigation menu list template used when you create a new list to be displayed on the side of the page.</td>
</tr>
<tr>
<td>Login Page</td>
<td>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.</td>
</tr>
<tr>
<td>Error Page</td>
<td>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.</td>
</tr>
<tr>
<td>Printer Friendly Page</td>
<td>Identifies the template to be used when the Application Express engine is in printer friendly mode.</td>
</tr>
<tr>
<td>Breadcrumb</td>
<td>Identifies the default breadcrumb template used when you create a breadcrumb.</td>
</tr>
<tr>
<td>Button</td>
<td>Identifies the default button template used when you create a button.</td>
</tr>
<tr>
<td>Legacy Calendar</td>
<td>Specifies the default calendar template used when you create a calendar.</td>
</tr>
<tr>
<td>Default Label</td>
<td>Specifies the default label template used when you create a new label.</td>
</tr>
<tr>
<td>Optional Label</td>
<td>Specifies the default optional label template used when you create new label.</td>
</tr>
<tr>
<td>Page Template Options</td>
<td>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.</td>
</tr>
<tr>
<td>Required Label</td>
<td>Specifies the default required label template used when you create a new label.</td>
</tr>
<tr>
<td>List</td>
<td>Specifies the default list template used when you create a list.</td>
</tr>
<tr>
<td>Region</td>
<td>Specifies the default region template used when you create a region.</td>
</tr>
<tr>
<td>Classic Report</td>
<td>Identifies the default region template used when you create a report.</td>
</tr>
</tbody>
</table>
Using Themes and Theme Styles

### Region Defaults
Table 13–4 describes the default templates available under Region Defaults.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breadcrumb</td>
<td>Identifies the default breadcrumb template used when you create a breadcrumb.</td>
</tr>
<tr>
<td>Charts</td>
<td>Specifies the default chart template used when you create a chart.</td>
</tr>
<tr>
<td>Forms</td>
<td>Specifies the default form template used when you create a form.</td>
</tr>
<tr>
<td>Lists</td>
<td>Specifies the default region template used when you create a list.</td>
</tr>
<tr>
<td>Reports</td>
<td>Specifies the default region template used when you create a report.</td>
</tr>
<tr>
<td>Tabular Forms</td>
<td>Specifies the default region template used when you create a tabular form.</td>
</tr>
<tr>
<td>Wizards</td>
<td>Specifies the default region template used when you create a new wizard component.</td>
</tr>
<tr>
<td>Interactive Reports</td>
<td>Specifies the default region template used when you create an interactive report</td>
</tr>
</tbody>
</table>

### Dialog Defaults
Table 13–5 describes the default templates available under Dialog Defaults.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialog Content Region</td>
<td>Specifies the default region template used when you create a content region on a Dialog using the Create Page Wizard and Create Application Wizard.</td>
</tr>
<tr>
<td>Dialog Button Region</td>
<td>Specifies the default region template used when you create a button region on a Dialog using the Create Page Wizard and Create Application Wizard.</td>
</tr>
<tr>
<td>Dialog Page</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Transition Defaults
Transition Defaults is only visible for a Mobile theme. Table 13–6 describes attributes available under Transition Defaults.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Transition</td>
<td>Specifies the transition which should be used when navigating to a new page or when the page is getting submitted.</td>
</tr>
</tbody>
</table>
Using Themes

**Table 13–6 (Cont.) Transition Defaults**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popup Transition</td>
<td>Specifies the transition which should be used when a popup is opened.</td>
</tr>
</tbody>
</table>

**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" on page 13-6 and "Managing Global Template Options" on page 13-27

**Icons** Table 13–7 describes attributes under Icons.

**Table 13–7 Icons**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>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.</td>
</tr>
<tr>
<td>Custom Library File URLs</td>
<td>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 #MIN# to include .min or #MIN_DIRECTORY# to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string #APP_VERSION# if you want to include the application's version in the file URL. File URLs you enter here will replace the #THEME_CSS# substitution string in the page template. To view examples, see field-level Help.</td>
</tr>
<tr>
<td>Custom Classes</td>
<td>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.</td>
</tr>
<tr>
<td>Custom Prefix Class</td>
<td>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.</td>
</tr>
<tr>
<td>Date Picker Icon Name</td>
<td>Enter the name and location of the image to be used when items of type Date Picker are displayed. For example: #IMAGE_PREFIX#new_cal.bmp</td>
</tr>
<tr>
<td>Date Picker Icon Attributes</td>
<td>Enter the image attributes for the Calendar Icon. For example: width=&quot;16&quot; height=&quot;16&quot; alt=&quot;Display Calendar&quot;</td>
</tr>
</tbody>
</table>

**Image** Use the Custom Image attribute to specify whether this theme uses a default theme image or an uploaded theme image. Select Yes or No.

**Styles** Displays theme styles available to the current theme. Theme Styles define the CSS for your application’s theme and control the user interface of your application. Theme styles are referenced in the page template using the #THEME_STYLE_CSS# substitution string. The Theme Style currently used displays a check mark under Is Current.
**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.

**Managing Themes and Subscriptions**

This section describes how developers can manage themes and theme subscriptions.

- **Understanding Theme Subscriptions**
- **Creating a Theme**
- **Switching Themes**
- **Changing a Theme Identification Number**
- **Copying a Theme**
- **Deleting a Theme**
- **Creating a Master Theme Application**
- **Managing Workspace Themes**
- **Managing Instance Themes**
- **About Exporting and Importing Themes**
- **Viewing Theme Reports**

**Understanding Theme Subscriptions**

This section describes theme subscriptions.

- **About Theme Subscriptions**
- **About Modifying a Theme Subscription**
- **Verifying if a Theme Refresh Is Needed**
- **Refreshing a Subscribed Theme**
- **Unsubscribing to a Theme**
- **Re-subscribing to a Theme**

**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" on page 7-1.
- **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" on page 13-15.
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.

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" on page 13-18

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:

2. Select the theme.
   The Create/Edit Theme page appears.
3. Locate Theme Subscription and click Verify.
4. Follow the on-screen instructions.
   To publish changes from the master theme to the subscribing theme, click Refresh at the end of the wizard.

Refreshing a Subscribed Theme

To refresh a subscribed theme:

2. Select the theme.
   The Create/Edit Theme page appears.
3. 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 Theme" on page 13-15.
4. Click Apply Changes.
Unsubscribing to a Theme

**Note:** Unsubscribing to a theme prevents future theme updates. Be aware that re-subscribing to a theme is multiple step process that involves copying and switching themes. To learn more, see "Re-subscribing to a Theme" on page 13-15.

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:
2. Select the theme.
   The Create/Edit Theme page appears.
3. Locate Theme Subscription and click **Unsubscribe**.
   A warning dialog displays.
4. Click **Unsubscribe Theme**.

Re-subscribing to a Theme

To re-subscribe to a theme.

2. Click **Change Identification Number** and change the theme to a new identification number. See "Changing a Theme Identification Number" on page 13-17.
3. Run the Create Theme Wizard and create the theme again. See "Creating a Theme" on page 13-15.
4. Switch the current application to the use the new theme. See "Switching Themes" on page 13-16.
5. Delete the original theme from which you created a new identification number. See "Deleting a Theme" on page 13-18.

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" on page 13-13.

To create a theme:
2. Click **Create**.
3. Select a creation method:
   - **From the Repository** - View and select a theme from the repository.
Managing Themes and Subscriptions

- **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 "Viewing Field-Level Help" on page 1-21.

**See Also:** "Editing Themes" on page 13-8

Switching Themes

This section describes how to change the active theme for an application.

- **About Switching the Active Theme**
- **About Switching to the Universal Theme**
- **Switching an Active Theme**

**About Switching the Active Theme**

When you switch to a theme, all components with assigned templates are assigned to a corresponding template in the theme. Application 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" on page 13-15.

- 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.

**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

See Also: "Editing Themes" on page 13-8
Switching an Active Theme

To switch an active theme:


   **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" on page 13-15.

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. See "About Grid-based Themes" on page 13-2.
   d. Match Template Classes - Select **Yes** or **No**. When switching to a new theme, Application 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:
   - **Check** indicates the mapping was successful.
   - **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.
   - **Error** indicates that Application Builder was unable to map the class among the themes. Ensure that a class is identified for the templates in both themes.

   **Note:** Failure to resolve warnings may result in a loss of functionality and applications that do not display well.

5. Click **Next** to continue.

6. Click **Switch Theme**.

Changing a Theme Identification Number

Each theme has an identification number (ID). You can use the Change Theme ID utility to change a theme ID to another identification number. Changing a theme ID is useful when you want to export a theme with a different number and then import it into another application.

To change a theme identification number:

2. On the Tasks list, click **Change Identification Number**.

3. For Theme Number:
   a. Identify Theme - Select a theme.
   b. New Theme Identification Number - Specify an identification number.
   c. Click **Next**.

4. Confirm your changes and click **Change Theme ID**.

### Copying a Theme

Each theme is identified by a numeric identification number (ID). When you copy a theme, you specify a theme ID. Copying a theme is useful when experimenting with editing a theme or to export a theme with a different ID.

To copy a theme:

2. On the Tasks list, click **Copy Theme**.
3. On Copy Theme:
   a. Copy From Theme - Select the theme you want to copy.
   b. Copy to this Theme ID - Enter a new ID for the theme.
   c. Subscribe Theme - Select **Yes** or **No**.
   d. Click **Next**.
4. Click **Copy Theme**.

### Deleting a Theme

You can only delete inactive themes. When you delete a theme, Application Builder only removes inactive templates.

**Tip:** Themes are tied to an application. If you modify a theme, those changes only apply to the application being edited. If you delete a theme, any template modifications are lost. To keep template modifications, you must export the theme. See "Exporting Themes" on page 24-19.

To delete a theme:

2. On the Tasks list, click **Delete Theme**.
3. From Delete Theme, select the theme you want to delete and click **Next**.
4. Click **Delete Theme**.

### Creating a Master Theme Application

This sections explains how to 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
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. See "Using the Create Application Wizard" on page 7-1.
2. Select **Copy from an existing application** and select an application that uses *Universal Theme - 42* (for example, Sample Database Application).
3. Follow the on-screen instructions.

Creating a Master Theme Application from an Existing Application

To create a master theme application from an existing application:

1. Copy the theme from an application that uses Universal Theme - 42 (for example, *Sample Database Application*) and subscribe to it. See "Copying a Theme" on page 13-18.
2. Switch to the new theme. See "Switching Themes" on page 13-16.
3. Follow the on-screen instructions.

Managing Workspace Themes

Workspace administrators manage the theme repository for a workspace. Workspace administrators can add a theme to the repository, making it available to all developers within a workspace, or delete it.

- Adding a Workspace Theme to the Theme Repository
- Viewing an Existing Workspace Theme
- Deleting a Workspace Theme
- Modifying a Workspace Theme
- Exporting a Workspace Theme

**See Also:** "Managing Public Themes" in Oracle Application Express Administration Guide

Adding a Workspace Theme to the Theme Repository

To add a theme to the Theme repository:

2. From Tasks, 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. Click **Next**.
5. Change the theme name and number:
   a. Theme Number - Enter a number with a value greater than 100 to uniquely identify this theme within a workspace.
   b. Theme Name - Enter a theme name. This name can be the same or different from the application theme name.
   c. Description - Enter a description of the theme.
   d. Click Next.

6. Click Create Workspace Theme.

Viewing an Existing Workspace Theme
To view an existing workspace theme:
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" on page 13-20.

Deleting a Workspace Theme
To delete a workspace theme:
2. From the Tasks List, click Manage Workspace Themes.
   The Workspace Themes page appears.
3. Select the theme name.
4. Click Delete.
5. Confirm your delete request and click OK.

Modifying a Workspace Theme
You cannot edit a workspace theme directly. To modify a workspace theme, you must create an application using the theme, modify it, and then manually add it to the workspace theme repository.

To modify a workspace theme:
1. Create an application using the theme you want to modify. "Creating a Database Application" on page 7-4.
3. Delete the existing workspace theme. See "Deleting a Workspace Theme" on page 13-20.
4. Add the modified theme to the workspace theme repository. See "Adding a Workspace Theme to the Theme Repository" on page 13-19.
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. "Creating a Database Application" on page 7-4.
2. Export the application. See "Exporting an Application" on page 24-14.
3. Import the exported file into the target Oracle Application Express instance. See "Importing an Application, Page or Component Export" on page 24-23.
4. Install the exported file from the Export Repository. See "Installing Export Files" on page 24-27.

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. To learn more, see "Managing Public Themes" in Oracle Application Express Administration Guide.

About Exporting and Importing Themes
You export a theme in the same way you export any related application files. Exporting a theme from one development instance to another involves the following steps:

1. Create an application using the theme. "Creating a Database Application" on page 7-4.
2. Export the theme. See "Exporting Themes" on page 24-19.
3. Import the exported file into the target Oracle Application Express instance. See "Importing Export Files" on page 24-23.
4. Install the exported file from the Export Repository. See "Installing Export Files" on page 24-27.

Viewing Theme Reports
This section describes several reports designed to help you manage themes and templates.

- Viewing All Templates in a Theme
- Viewing Theme Template Counts
- Viewing File References
- Viewing Class References
- Viewing Template Substitution Strings

Viewing All Templates in a Theme
To view all templates that comprise a theme:

2. Click Reports.
3. On the Theme Reports page:
Managing Themes and Subscriptions

1. Report - Select Application Templates.
2. Theme - Select a theme or select All Themes.
3. 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.

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:
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.

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:
2. On the Themes page, click Reports.
3. On the Theme Reports page:
   b. Theme - Select a theme or select All Themes.
   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.
**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.

---

**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:

2. Click Reports.
3. On the Theme Reports page:
   a. Report - Select **Class References**.
   b. Theme - Select a theme or select **All Themes**.
   c. Click Go.
4. On the Class References page:
   a. Show - Select the components to check for a class reference. If you do not make a selection, no results are returned.
   b. Show Class Names - Select one of the following:
      - **With context** displays the component, the theme identification number, the component name, the image (if applicable), and the page number.
      - **Without context** displays only the referenced class.
   c. Click Go.
5. To download a comma-delimited file (.csv) version of this report, click **Download** at the bottom of the page.

**Viewing Template Substitution Strings**

Use the Template Substitution Strings report to view all supported substitution strings by component.

To view the Substitution String report:

2. Click Reports.
3. On the Theme Reports page:
   a. Report list - Select **Template Substitution Strings**.
   b. Theme list - 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" on page 2-25
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" on page 13-58

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
Creating Custom Themes

Using Themes and Theme Styles

13-25

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" on page 13-43

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" on page 2-26

Selecting a Default Page Template

This section describes how to select a default page template. You can specify a default page template in two ways:

Select a default page template within a specific theme.
Select a specific page template on a page-by-page basis.

By default, the Application Express engine uses the Page template specified on the Themes page.

Selecting a Page-level Template Within a Theme

Selecting a Page-level Template for a Specific Page

Selecting a Page-level Template Within a Theme

To specify a default page template within a theme:

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" on page 13-8

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 Application Builder icon.
2. Select an application.
3. Select a page.
   The page appears.
4. To edit the page-level template in Page Designer:
From the Page Rendering tree, click the page title.

b. In the Property Editor, locate the Appearance section.
c. Under Appearance, Page Template, select a new template.
d. Click Save.

5. To edit the page-level template in Component View:

a. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
b. Under Page, click the Edit icon.
   The Edit page appears.
c. Under Display Appearance, Page Template, select a new template.
d. Click Apply Changes.

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 Application Builder icon.
2. Select the application to which you want to copy the template.
3. Click Shared Components.
4. Under User Interface, select Templates.
5. Click Create.
6. Select the type of template you want to create and click Next.
7. Select a creation method:
   ■ From Scratch
   ■ As a Copy of an Existing Template
8. Follow the on-screen instructions.

Tip: Make sure you associate your template with the correct theme.

Using Template Options

Template options provide a declarative way for developers to apply different styles to components on an Oracle Application Express page.

■ Managing Global Template Option Groups
■ Managing Global Template Options
■ Managing Template Options

See Also: "Understanding Template Options" on page 13-6
Managing Global Template Option Groups
Global template option groups are defined at the theme-level and are available for all components of a given type. This section describes how to create and edit global template option groups.

- Creating a Global Template Option Group
- Editing Global Template Option Groups

See Also: "About Template Option Groups and Presets" on page 13-7 and "Understanding Template Options" on page 13-6

Creating a Global Template Option Group To create a global template option group:
2. Select a theme.
   The Create/Edit Theme page appears.
3. Select Global Template Options.
4. Click Edit Template Option Groups.
   Template Options Groups dialog appears.
5. To create a group:
   a. Click Create.
   b. Edit the appropriate attributes.
      To learn more about an attribute, see field-level Help.
   c. Click Create.

Editing Global Template Option Groups To edit global template option groups:
2. Select a theme.
   The Create/Edit Theme page appears.
3. Select 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.
   c. Click Apply Changes.
6. To delete a group:
   a. Click the group name.
   b. Click Delete.

Managing Global Template Options
This section describes how to create and edit global template options.

- Creating Global Template Options
Creating Custom Themes

Editing Global Template Options

See Also: "About Template Options" on page 13-6

Creating Global Template Options To create a global template option:

2. Select a theme.
   The Create/Edit Theme page appears.
3. Select Global Template Options.
4. Click Add Template Option.
5. On Template Options:
   a. Display Sequence - Specify the sequence for this option.
   b. Template Type - Select the type of template for which the template option is defined.
   c. Group - Enter a unique group name. Group names enable you to group template options and avoid the selection of conflicting option.
      Display Name - Enter a name for the template option.
   d. Option Identifier - Enter an alphanumeric, upper case identifier for the template option. Identifiers map a template option to a corresponding template option when switching themes.
      CSS Classes - Enter one or more CSS classes that define this template option.
      Help Text - Enter a description of the template option.
      Click Create or Create and Add Another.

Editing Global Template Options To edit a global template option:

2. Select a theme.
   The Create/Edit Theme page appears.
3. Select Global Template Options.
4. To edit a template option:
   a. Click the display name.
   b. Edit the appropriate attributes. To learn more about an attribute, see field-level Help.

   Tip: If you edit the name of a CSS class, all components that reference that CSS class name must be updated manually. The Option Identifier cannot be modified since it is needed for mapping purposes.
   c. Click Apply Changes.
5. To delete a template option:
   a. Click the display name.
   b. Click Delete.
Managing Template Options
This section describes how to create and edit template options.

- Creating Template Options
- Editing Template Options

See Also: "Understanding Template Options" on page 13-6

Creating Template Options To create a template option:
1. Navigate to the Templates page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
   d. Under User Interface, select Templates.
2. Select a template.
3. Select Template Options.
4. Click Add Template Option.
5. Edit the appropriate attributes.
   To learn more about an attribute, see field-level Help.
6. Click Create or Create and Add Another.

Editing Template Options To edit a template option:
1. Navigate to the Templates page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
   d. Under User Interface, select Templates.
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:
   a. Click Delete.
6. To view global template options, expand Global Template Options.

Viewing Template Reports
Application 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 **Application 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. To learn more, see “Customizing Interactive Reports as an End User” on page 10-57.

5. To view template reports, click the following tabs:
   - **Subscription** - Displays subscribed templates in your application.
   - **Publish** - displays templates that have been subscribed to by other templates in your workspace. To publish master template content to subscribing templates, select those you want to publish and click Publish Checked.
   - **Utilization** - Displays template utilization in the current application for all template types. The number in the References column indicates the total number of pages that use the specific template.
   - **History** - Displays recent modifications made to Page and Region Templates in this application.

**Managing Templates**

You can view all available templates on the Templates page. Alternatively, you can access a template associated with a specific page.

- Viewing Templates on the Templates Page
- Viewing Templates Associated with a Specific Page
- Replacing Templates
- Replacing All Templates within an Application
- Viewing Region Position Utilization by Page Template
- Unsubscribing to Templates
- Publishing Templates

**See Also:**  "Viewing All Templates in a Theme” on page 13-21

**Viewing Templates on the Templates Page**

To view existing templates:

1. On the Workspace home page, click the **Application 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. To learn more, see “Customizing Interactive Reports as an End User” on page 10-57.

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. See “Viewing Field-Level Help” on page 1-21.

If you edit a template, you can make changes in one window and run your application in another by selecting the **Return to Page** check box on the right side of the template definition page. Selecting this check box keeps the page you are editing current after you click Apply Changes.

7. Click Apply Changes.

**Viewing Templates Associated with a Specific Page**

To view templates associated with a specific page:

1. Navigate to the appropriate page. See "Viewing a Page in Component View” on page 9-5.

   The page appears. Templates associated with the current page display under the Templates heading in the far right column.

2. To view or edit a template definition, click the template name.

   The template definition appears.

3. Edit the appropriate attributes. To learn more about an attribute, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

   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.

4. Click Apply Changes.

**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 **Application 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. To learn more, see "Customizing Interactive Reports as an End User” on page 10-57.

5. From the Task list, select **Replace Templates**.

6. From Template Type, select a template and click **Next**.

7. For Replace Templates:

   a. Change From - Select the template you want to change.

   b. Change To - Select the template you want to change to.

   c. Click **Next**.

8. Click Finish.
Replacing All Templates within an Application
To replace all templates within an application with templates from another application:

1. On the Workspace home page, click the **Application 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. To learn more, see "Customizing Interactive Reports as an End User" on page 10-57.
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:
   - **Replace** copies the template definition
   - **Replace/Subscribe** copies the templates and adds a subscription.
8. Click **Replace Templates**.

Viewing Region Position Utilization by Page Template
Regions are organized on a page by position (or Display Point). The possible display points for a region are determined by the page-level template.

To view region position utilization by page template:

1. On the Workspace home page, click the **Application 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. To learn more, see "Customizing Interactive Reports as an End User" on page 10-57.
5. From the Task list, select **View page template region position utilization**.
   The Region Display Point Utilization report appears.

Unsubscribing to Templates
A subscribed template is a template that has its definition maintained in another template, the referenced template. If your application uses subscribed templates, you can unsubscribe to templates on the Unsubscribe Templates page.

To unsubscribe to templates:
1. On the Workspace home page, click the Application 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. To learn more, see "Customizing Interactive Reports as an End User" on page 10-57.

5. From the Task list, select Unsubscribe Templates.
   The Unsubscribe Templates page appears.

6. To unsubscribe to a specific template type within the current theme, select the template type and click Unsubscribe.

**Publishing Templates**

Use the Publish Templates page to view templates that are subscribed to by other templates in your workspace.

To publish master template content to subscribing templates:

1. On the Workspace home page, click the Application 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. To learn more, see "Customizing Interactive Reports as an End User" on page 10-57.

5. Click the Publish tab.
   The Published Templates page appears.

6. Select those you want to publish and click Publish Checked.

**Breadcrumb Templates**

A breadcrumb template controls the display of breadcrumb entries. You select a breadcrumb template when you create a region.

- About Breadcrumb Style Navigation
- Breadcrumb Template Attributes

See also: "Using Themes" on page 13-1

**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.
Creating Custom Themes

**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**  
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. Application Builder accomplishes this template mapping through the assignment of a template class.

**Definition**  
Definition attributes control how a breadcrumb displays. Table 13–8 describes Definition attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before First</td>
<td>Defines text that displays before the first breadcrumb entry. Use the substitution string #COMPONENT_CSS_CLASSES# to apply a CSS class to your breadcrumb template. You edit component CSS classes by editing the region attributes. <strong>See Also:</strong> &quot;Editing Region Attributes in Page Designer&quot; on page 14-5 and &quot;Editing Region Attributes in Component View&quot; on page 14-15</td>
</tr>
<tr>
<td>Current Page Breadcrumb Entry</td>
<td>Defines the look of a breadcrumb entry that corresponds to the current page.</td>
</tr>
<tr>
<td>Non Current Page Breadcrumb Entry</td>
<td>Defines the look of a breadcrumb entry that does not correspond to the current page.</td>
</tr>
<tr>
<td>After Last</td>
<td>Defines text that displays after the last breadcrumb entry.</td>
</tr>
<tr>
<td>Between Level</td>
<td>Defines text that displays between levels of breadcrumb entries.</td>
</tr>
</tbody>
</table>
Creating Custom Themes

Using Themes and Theme Styles

13-35

Substitution Strings

Substitution strings are used within subtemplates to reference component values. Table 13–9 describes the available button template substitution strings.

**Tip:** All template substitution strings must be in uppercase letters and begin and end with a number sign (#). To view a report of substitution strings supported by a given template, see the Substitution Strings section of the Edit Breadcrumb Template page.

Table 13–9 Breadcrumb Template Substitution Strings

<table>
<thead>
<tr>
<th>Referenced From</th>
<th>Substitution String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before First</td>
<td>#COMPONENT_CSS_CLASSES#</td>
<td>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.</td>
</tr>
<tr>
<td>Current Entry</td>
<td>#LINK#</td>
<td>URL Link.</td>
</tr>
<tr>
<td>Current Entry</td>
<td>#LONG_NAME#</td>
<td>Breadcrumb Long Name.</td>
</tr>
<tr>
<td>Current Entry</td>
<td>#NAME#</td>
<td>Breadcrumb Label.</td>
</tr>
<tr>
<td>Non Current Entry</td>
<td>#NAME_ESC_SC#</td>
<td>Breadcrumb Label (escaping HTML characters).</td>
</tr>
<tr>
<td>Non Current Entry</td>
<td>#LINK#</td>
<td>URL Link.</td>
</tr>
<tr>
<td>Non Current Entry</td>
<td>#LONG_NAME#</td>
<td>Breadcrumb Long Name.</td>
</tr>
<tr>
<td>Non Current Entry</td>
<td>#NAME#</td>
<td>Breadcrumb Label.</td>
</tr>
<tr>
<td>Non Current Entry</td>
<td>#NAME_ESC_SC#</td>
<td>Breadcrumb Label (escaping HTML characters).</td>
</tr>
</tbody>
</table>

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" on page 13-1

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" on page 13-6 and "Understanding Theme Subscriptions" on page 13-13

**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. Application Builder accomplishes this template mapping through the assignment of a template class.

Definition Normal Template defines the button template that displays if the button type is Normal. For example, &ITEM_NAME. values can be substituted at rendering time. Hot Template defines the button template that displays if the button type is Hot. You have the option of including standard application substitutions. For example, &ITEM_NAME. values can be substituted at rendering time.

To learn more about supported substitution strings, see field-level Help.

Tip: All substitution string must be in uppercase.

Substitution Strings Substitution strings are used within subtemplates to reference component values. Table 13–10 describes the available button template substitution strings.

Tip: All template substitution strings must be in uppercase letters and begin and end with a number sign (#). To view a report of substitution strings supported by a given template, see the Substitution Stings section of the Edit Page Template page.

<table>
<thead>
<tr>
<th>Referenced From</th>
<th>Substitution String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hot Template</td>
<td>&amp;BUTTON_ATTRIBUTES#</td>
<td>Button attributes.</td>
</tr>
<tr>
<td>Hot Template</td>
<td>&amp;BUTTON_CSS_CLASSES#</td>
<td>Button CSS classes. You must include the &amp;BUTTON_CSS_CLASSES# string for templates that have template options defined. The &amp;BUTTON_CSS_CLASSES# string is where the corresponding template option classes are injected at runtime.</td>
</tr>
<tr>
<td>Hot Template</td>
<td>&amp;BUTTON_ID#</td>
<td>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'</td>
</tr>
<tr>
<td>Hot Template</td>
<td>&amp;ICON_CSS_CLASSES#</td>
<td>Icon CSS classes.</td>
</tr>
<tr>
<td>Hot Template</td>
<td>&amp;JAVASCRIPT#</td>
<td>Used in an onclick attribute.</td>
</tr>
<tr>
<td>Hot Template</td>
<td>&amp;LABEL#</td>
<td>Button label.</td>
</tr>
<tr>
<td>Hot Template</td>
<td>&amp;LINK#</td>
<td>To be used in a href attribute.</td>
</tr>
<tr>
<td>Template</td>
<td>&amp;BUTTON_ATTRIBUTES#</td>
<td>Button attributes.</td>
</tr>
<tr>
<td>Template</td>
<td>&amp;BUTTON_CSS_CLASSES#</td>
<td>Button CSS classes. You must include the &amp;BUTTON_CSS_CLASSES# string for templates that have template options defined. The &amp;BUTTON_CSS_CLASSES# string is where the corresponding template option classes are injected at runtime.</td>
</tr>
<tr>
<td>Template</td>
<td>&amp;BUTTON_ID#</td>
<td>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'</td>
</tr>
<tr>
<td>Template</td>
<td>&amp;ICON_CSS_CLASSES#</td>
<td>Icon CSS classes.</td>
</tr>
</tbody>
</table>
Creating Custom Themes

Using Themes and Theme Styles

13-37

Legacy Calendar Templates

Legacy Calendar templates control the appearance and placement of a calendar. Calendar templates frequently use HTML tables to arrange dates. You place calendar attributes using substitution strings such as #DD# and #MONTH#.

- Supported Legacy Calendar Template Substitution Strings
- Legacy Calendar Template Attributes

See Also: "About Supported Calendars" on page 12-1, "Creating Calendars" on page 12-1, and "Using Themes" on page 13-1

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 (#).

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" on page 13-6 and "Understanding Theme Subscriptions" on page 13-13

Calendar Definition

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. Application Builder accomplishes this template mapping through the assignment of a template class.

Subscription Use Subscription to apply an existing template to the current application. Use Reference Master Label Template From to select an existing template in this workspace or another template in your workspace’s schema. By selecting an existing template, you become a subscriber to that template.

To load a copy of a master template, click Refresh.

Comments Use this attribute to record comments about this component.

<table>
<thead>
<tr>
<th>Referenced From</th>
<th>Substitution String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template</td>
<td>#JAVA#SCRIPT#</td>
<td>Used in an onclick attribute.</td>
</tr>
<tr>
<td>Template</td>
<td>#LABEL#</td>
<td>Button label.</td>
</tr>
<tr>
<td>Template</td>
<td>#LINK#</td>
<td>To be used in a href attribute.</td>
</tr>
</tbody>
</table>
Creating Custom Themes

**Monthly Calendar, Weekly Calendar, Daily Calendar, Custom Calendar and List Calendar**

Use the Monthly Calendar, Weekly Calendar, Daily Calendar, and List Calendar attributes to control the appearance and placement of specific calendars.

**Label Templates**

Label templates are designed to centrally manage HTML markup of page item labels. Each item can have an optional label. You can control how these labels display using label templates. For example, you could create a label template called Required Field that references an image (such as an asterisk) to indicate to the user that the field is required.

Label templates enable you to define a before-and-after text string that gets prepended and appended to the item.

- **Label Template Attributes**
- **About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help**

**See Also:** "Using Themes" on page 13-1

**Label Template Attributes**

**See Also:** "Understanding Template Options" on page 13-6 and "Understanding Theme Subscriptions" on page 13-13

This section describes specific sections of the Edit Label Template page. To learn more about a specific attribute, see field-level Help.

**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. Application Builder accomplishes this template mapping through the assignment of a template class.

**Definition** Definition attributes include:

**Tip:** For a list of supported substitution strings and to view examples, see field-level Help.

- **Before Label** - Enter HTML to display before the item label. The #ITEM_CSS_CLASSES# substitution string must be included for templates that have template options defined.
- **After Label** - Enter HTML to display after the item label. Since the label automatically displays before the HTML in this region, any open HTML tags in the Before Label region should be closed.
- **Before Item** - Enter HTML to be displayed directly before the form item. This is typically used to wrap an item into a DIV container or to display a help button before or after the form item.
Creating Custom Themes

Using Themes and Theme Styles

- **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.

- **Help Template** - The help sub template allows for defining 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" on page 13-39.

**Error Display**  Error Display attributes include:

  **Tip:**  For a list of supported substitution strings and to view examples, see field-level Help.

- **On Error Before Label** - Enter HTML to precede the item label when an application displays an inline validation error message for the item.

- **On Error After Label** - Enter HTML to be appended to the item label when an application displays an inline validation error message for the item.

- **Error Template** - Enter the HTML rendered to display the inline validation error message. Use the #ERROR_MESSAGE# substitution string to reference the message text. The error template can be included in the before and after item attributes using the #ERROR_TEMPLATE# substitution string.

**Field Container**  Field Container attributes include:

  **Tip:**  For a list of supported substitution strings and to view examples, see field-level Help.

- **Before Label and Item** - Enter HTML to be displayed before the label and item. This is typically used to wrap an item into a DIV container or field set. The Before Label and Item attribute is only rendered for pages that do not use a table grid to display form items. This includes any page using a mobile page template.

- **After Label and Item** - Enter HTML to be displayed after the label and item. This is typically used to wrap an item into a DIV container or field set. The after item and label attribute is only rendered for pages that don't use a table grid to display form items. This includes any page using a mobile page template.

**Substitution Strings** Lists substitution string usage for this template. Substitution strings are used within subtemplates to reference component values.

**About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help**

You can use the #CURRENT_ITEM_HELP_TEXT# substitution string to create inline page Help text when a page is rendered. For example, you can use this substitution string to create page Help text which displays as a tooltip when the mouse hovers over the page item label. This placeholder enables you to include help text on the page when it is rendered without having to perform an expensive Ajax call.

  **Tip:**  Only use this placeholder if the Help text is short. The generated page always includes the help text of all page items.

**Example Label Template**

Before Label:

```html
<label for="#CURRENT_ITEM_NAME#" id="#LABEL_ID#">
```

Tip: For a list of supported substitution strings and to view examples, see field-level Help.
Create Custom Themes

List Templates

A list is a shared collection of links. You control the appearance of a list through list templates. Using template attributes, you can also define a list element to be either current or non current for a specific page.

- About Hierarchical Lists
- List Template Attributes

See Also: "Creating Lists" on page 18-20 and "Using Themes" on page 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" on page 18-20

List Template Attributes

See Also: "Understanding Template Options" on page 13-6 and "Understanding Theme Subscriptions" on page 13-13

This section describes some sections of the Edit List Template page. To learn more about a specific attribute, see field-level Help.

**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. Application Builder accomplishes this template mapping through the assignment of a template class.

**Before List Entry**

Enter HTML that displays before any list elements. You can use this attribute to open an HTML table or HTML table row. The #COMPONENT_CSS_CLASSES# substitution string must be included in the List Template Before Rows attribute for templates that have template options defined.

**Template Definition**

Defines current and noncurrent list templates. Supported substitution strings include #LINK#, #TEXT#, #TEXT_ESC_SC#, #ICON_CSS_CLASSES#, #IMAGE_PREFIX#, #IMAGE#, #IMAGE_ATTR#, and #A01#...#A10#.
Template Definition attributes include:

- **List Template Current.** Enter HTML or text to be substituted for the selected (or current) list template.

- **List Template Current with Sub List Items.** Enter HTML or text to be substituted for the selected (or current) list template when an item has sublist items. If not specified, the current list item template is used.

- **List Template Noncurrent.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template.

- **List Template Noncurrent with Sub List Items.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template used when an item has sublist items. If not specified, the current list item template is used.

- **Between List Elements.** Enter HTML that displays between list elements. This attribute is ignored if no HTML is specified.

**Template Definitions for First Entry**  
Available attributes include:

- **List Template Current (First).** If defined, this is used for the first list item. Enter HTML or text to be substituted for the selected (or current) list template.

- **List Template Current with Sub List Items (First).** If defined, this is used for the first list item. HTML or text to be substituted for the selected (or current) list template used when item has sublist 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 sublist items.

**Before Sublist Entry**  
Enter HTML that displays before any sublist elements.

**Sublist Entry**  
Defines current and noncurrent list templates. Supported substitution strings include #LINK#, #TEXT#, #ICON_CSS_CLASSES#, #IMAGE_PREFIX#, #IMAGE#, #IMAGE_ATTR#, #ICON_CSS_CLASSES#, #A01#...#A10#, #LIST_ITEM_ID#, and #PARENT_LIST_ITEM_ID#.

Sub List Entry attributes include:

- **Sub List Template Current.** Enter HTML or text to be substituted for the selected (or current) list template.

- **Sub List Template Current with Sub List Items.** Enter HTML or text to be substituted for the selected (or current) list template when an item has sublist items. If not specified, the current list item template is used.

- **Sub List Template Noncurrent.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template.

- **Sub List Template Noncurrent with Sub List Items.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template used when an item has sublist items. If not specified, the current list item template is used.

- **Between Sub List Items.** Enter HTML that displays between list elements. This attribute is ignored if no HTML is specified.

**After Sub List Entry**  
Enter HTML that displays after displaying sublist elements.
After List Entry  Enter HTML that displays after displaying all list elements. You can use this attribute to close an HTML table opened in the Before List Elements attribute.

Substitution Strings  Lists substitution string usage for this template. Substitution strings are used within subtemplates to reference component values.

Page Templates

Page templates define the appearance of a page. Each template consists of a header template, a body template, a footer template, and several subtemplates. If you do not specify a page template as a page-level attribute, then the Application Express engine uses the default page template defined on the Create/Edit Theme page.

Page templates combine static HTML with substitution strings that are replaced at runtime. You use substitution strings to indicate the existence and placement of a component within a page template. You can further specify how a component should display using subtemplates.

- Page Template Attributes
  - Required Dialog Page Template Attributes

See Also:  "Using Themes" on page 13-1

Page Template Attributes

See Also:  "Understanding Template Options" on page 13-6 and "Understanding Theme Subscriptions" on page 13-13

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  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. Application 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" on page 7-40 and "Required Dialog Page Template Attributes" on page 13-48

Definition  Each template consists of a header, a body, a footer, and subtemplates. Use substitution strings to include dynamic content. All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

Tip:  For a list of supported substitution strings and to view examples, see field-level Help.
Definition attributes include:

- **Header** - Enter HTML that defines the `<Head>` section of the HTML document. Oracle recommends you include at least `#TITLE#` and `#HEAD#` in this template section.
- **Body** - Enter HTML that makes up the `<Body>` section of the HTML document. Include substitution strings to define where the body content is placed and to define additional positions for regions.
- **Footer** - Enter HTML to define third section in the page template that displays after the Body.

**JavaScript** Available attributes include:

  **Tip:** For a list of supported substitution strings and to view examples, see field-level Help.

- **File URLs** - Enter JavaScript file URLs for code to be loaded on this page. Each URL must be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include .min in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You can also use the substitution string `#APP_VERSION#` to include the application's version in the file URL. JavaScript file URLs you enter here replaces the `#TEMPLATE_JAVASCRIPT#` substitution string in the page template.

  Do not include opening or closing script tags, just write the URL.

- **Function and Global Variable Declaration** - Enter JavaScript code (for example, functions or global variable declarations) for code to be used in this page template. Consider putting it into an external file to avoid duplication and to leverage browser caching of static files. Code you enter here replaces the `#TEMPLATE_JAVASCRIPT#` substitution string in the page template.

  Do not include opening or closing script tags, just include the JavaScript code.

  To reference a shortcut, use "SHORTCUTNAME".

- **Execute when Page Loads** - Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle Application Express. Code you enter here replaces the `#GENERATED_JAVASCRIPT#` substitution string in the page template.

- **Dialog Initialization Code** - This attribute only displays if the Template Type is Dialog Page. Enter JavaScript code to initialize the page as a dialog. The substitution strings will be replaced with their respective values at rendering time.

- **Dialog Closure Code**. This attribute only displays if the Template Type is Dialog Page. Enter JavaScript code to close the dialog page.

- **Dialog Cancel Code**. This attribute only applies if the Template Type is Dialog Page. Enter JavaScript code to cancel the dialog page.

**Cascading Style Sheet** Available attributes include:

- **File URLs** - Enter Cascading Style Sheet file URLs to be loaded for this page template. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include .min in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You can also use the substitution string `#APP_VERSION#` to include the
application’s version in the file URL. File URLs you enter here replace the #TEMPLATE_CSS# substitution string in the template.

Do not include opening or closing script tags, just include the file URL.

- **Inline** - Enter Cascading Style Sheet code to be used for this page template. Consider putting it into an external file to avoid duplication and to leverage browser caching of static files. Code you enter here replaces the #TEMPLATE_CSS# substitution string in the template.

Do not include opening or closing style tags, just include the cascading style sheet code.

**Subtemplate** Enter the HTML for subtemplates below. Subtemplates are placed inside the body of the page template using substitution strings.

Subtemplates include:

- **Success Message** - Enter HTML to replace the string #SUCCESS_MESSAGE# in the template body, header, or footer.

- **Navigation Bar** - Enter HTML or text to replace the string #NAVIGATION_BAR# in the template header, body, or footer. Use the substitution string #BAR_BODY# to indicate where in the Navigation Bar the entries should appear.

- **Navigation Bar Entry** - Enter HTML or text to be substituted into the navigation bar #BAR_BODY# substitution string for each navigation bar entry.

- **Notification** - Enter HTML or text to be substituted when the #NOTIFICATION_MESSAGE# substitution string is referenced in the template header, body, or footer.

**Image Based Tab Attributes** Use this subtemplate for tabs that are entirely based on images. Available attributes include:

- **Current Image Tab** - Enter HTML to be used to indicate that an image-based tab is currently selected. Include the #TAB_TEXT# substitution string to show the displayed name of the tab.

- **Non Current Image Tab** - Enter the HTML to be used to indicate that an image tab is not currently selected. Include the #TAB_TEXT# substitution string to show the displayed name of the tab.

**Grid Layout** Specified the type of layout grid 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 grid layout:

- **HTML Table** - Uses HTML Tables to position regions, page items, and buttons.

- **Fixed Number of Columns** - Used for grids which divide the screen into a fixed number of grid columns.

- **Variable Number of Columns** - Used for grids which can display up to "n" number of grid columns depending on how the regions, page items and buttons are positioned.

**Display Points** Available attributes include:

- **Breadcrumb Display Point** - Applies to generated components that use breadcrumbs. It defines where the breadcrumbs are placed on the page. Sidebar Display Point
- **Sidebar Display Point** - Applies to generated components that use Sidebars. It defines where sidebars are placed on the page.

**Dialog** Dialog attributes only appear if you select a Template Type of **Dialog Page**. These settings are overwritten by a similar attributes specified at page-level.

Available Dialog attributes include:

- **Width** - Enter the width of the dialog, in pixels or a percentage.
- **Height** - Enter the height of the dialog in pixels, for example, 500.
- **Maximum Width** - Enter the maximum width of the dialog, in pixels. For a page using the jQuery Mobile Smartphone user interface, the maximum width is 500.
- **CSS Classes** - Enter additional CSS classes you wish to be applied to the dialog, as a space delimited list.
- **Allow Embed in Frames** - Your page template level browser frame options are listed below, along with the current setting for each. Make a selection to control if the browser is allowed to display your application's pages within a frame. The current setting will be the default for each modal and non-modal page using this page template. To learn more, see field-level Help.

Oracle Application Express embeds modal dialogs in iframes. If you set the application security attribute Browser Frame to **Deny**, checking **Allow Embed In Frame** for Modal Dialog will instead use **Allow from same origin** for dialog pages that use this template.

**See Also:** "Required Dialog Page Template Attributes" on page 13-48

**Error Page Template Control** Use this attribute when a page template is designated as an error template.

**Substitution Strings** Table 13–11 describes the available page template substitution strings. Substitution strings are used within subtemplates to reference component values.

- **Tip:** All template substitution strings must be in uppercase letters and begin and end with a number sign (#). To view a report of substitution strings supported by a given template, see the Substitution Stings section of the Edit Page Template page.

<table>
<thead>
<tr>
<th>Referenced From</th>
<th>Substitution String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>#HIGH_CONTRAST_TOGGLE#</td>
<td>High Contrast toggle.</td>
</tr>
<tr>
<td>All</td>
<td>#PAGE_STATIC_ID#</td>
<td>Page Static ID (for example 'P100')</td>
</tr>
<tr>
<td>All</td>
<td>#PARENT_TAB_CELLS#</td>
<td>Display of parent tabs.</td>
</tr>
<tr>
<td>All</td>
<td>#REGION_POSITION_01#</td>
<td>Region Position 1.</td>
</tr>
<tr>
<td>All</td>
<td>#REGION_POSITION_02#</td>
<td>Region Position 2.</td>
</tr>
<tr>
<td>All</td>
<td>#REGION_POSITION_03#</td>
<td>Region Position 3.</td>
</tr>
<tr>
<td>All</td>
<td>#REGION_POSITION_04#</td>
<td>Region Position 4.</td>
</tr>
<tr>
<td>All</td>
<td>#REGION_POSITION_05#</td>
<td>Region Position 5.</td>
</tr>
<tr>
<td>All</td>
<td>#REGION_POSITION_06#</td>
<td>Region Position 6.</td>
</tr>
</tbody>
</table>
### Table 13–11 (Cont.) Page Template Substitution Strings

<table>
<thead>
<tr>
<th>Referenced From</th>
<th>Substitution String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>#REGION_POSITION_07#</td>
<td>Region Position 7.</td>
</tr>
<tr>
<td>All</td>
<td>#REGION_POSITION_08#</td>
<td>Region Position 8.</td>
</tr>
<tr>
<td>All</td>
<td>#REQUESTED_URL#</td>
<td>Page URL requested by the browser</td>
</tr>
<tr>
<td>All</td>
<td>#SCREEN_READER_TOGGLE#</td>
<td>Screen Reader toggle.</td>
</tr>
<tr>
<td>All</td>
<td>#SWITCH_UI_TO_DESKTOP#</td>
<td>Hyperlink for home page of desktop user interface.</td>
</tr>
<tr>
<td>All</td>
<td>#SWITCH_UI_TO_JQM_SMARTPHONE#</td>
<td>Hyperlink for home page of smart phone user interface.</td>
</tr>
<tr>
<td>All</td>
<td>#TITLE#</td>
<td>Defines the page title. Typically included within HTML title tags.</td>
</tr>
<tr>
<td>Body</td>
<td>#BODY#</td>
<td>Page Body.</td>
</tr>
<tr>
<td>Body</td>
<td>#BOX_BODY#</td>
<td>Where the Body displays. If the Body is null, then #BOX_BODY# is used instead.</td>
</tr>
<tr>
<td>Body</td>
<td>#FORM_OPEN#</td>
<td>Where the HTML open form tag <code>&lt;form&gt;</code> is placed. You must include this substitution string to submit a form.</td>
</tr>
<tr>
<td>Body</td>
<td>#GENERATED_CSS#</td>
<td>Cascading style sheet files created by components.</td>
</tr>
</tbody>
</table>
| 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_APPLICATIONS_GLOBAL_NOTIFICATION` to set this value programmatically.  
**See Also:** "Global Notification" on page 7-18 |
| 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" on page 7-31 |
| Body            | #NAVIGATION_BAR#    | Navigation bar.  
**See Also:** "Subtemplate" on page 13-44 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            | #SUCCESS_MESSAGE#   | Success message. Defines where page success and error messages appear. If the page process runs without raising errors, then this text displays.  
You can customize the display of the success message for each template by adding HTML to be displayed before and after the success message. |
| Body            | #TAB_CELLS#         | Display of standard tabs. |
| Body            | #TEMPLATE_JAVASCRIPT# | Display of standard tabs. |
| Footer          | #CUSTOMIZE#         | Customize link. |
| Footer          | #FORM_CLOSE#        | Where the HTML close form tag `<form>` is placed. |
### Table 13–11 (Cont.) Page Template Substitution Strings

<table>
<thead>
<tr>
<th>Referenced From</th>
<th>Substitution String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>#APEX_CSS#</td>
<td>Cascading Style Sheet files used by Application Express.</td>
</tr>
<tr>
<td>Header</td>
<td>#APEX_JAVASCRIPT#</td>
<td>JavaScript files and code used by Oracle Application Express.</td>
</tr>
<tr>
<td>Header</td>
<td>#APPLICATION_CSS#</td>
<td>CSS files defined on user interface level.</td>
</tr>
<tr>
<td>Header</td>
<td>#APPLICATION_JAVASCRIPT#</td>
<td>JavaScript files defined on user interface level.</td>
</tr>
<tr>
<td>Header</td>
<td>#GENERATED_JAVASCRIPT#</td>
<td>JavaScript code created by components as needed.</td>
</tr>
<tr>
<td>Header</td>
<td>#HEAD#</td>
<td>Used after the <code>&lt;head&gt;</code> open tag but before the <code>&lt;/head&gt;</code> close tag. You can optionally define the contents of <code>#HEAD#</code> for each page (for example, to reference additional style sheets or JavaScript libraries).</td>
</tr>
<tr>
<td>Header</td>
<td>#ONLOAD#</td>
<td>Page onload.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Header</td>
<td>#PAGE_CSS#</td>
<td>Cascading style sheet files and inline style defined on the page-level.</td>
</tr>
<tr>
<td>Header</td>
<td>#PAGE_JAVASCRIPT#</td>
<td>JavaScript files and code defined on the page-level.</td>
</tr>
<tr>
<td>Header</td>
<td>#TEMPLATE_CSS#</td>
<td>Cascading style sheet files and inline style defined on page template-level.</td>
</tr>
<tr>
<td>Header</td>
<td>#THEME_CSS#</td>
<td>Cascading Style Sheet files defined at the theme-level.</td>
</tr>
<tr>
<td>Header</td>
<td>#THEME_JAVASCRIPT#</td>
<td>JavaScript files defined on theme-level.</td>
</tr>
<tr>
<td>Header</td>
<td>#THEME_STYLE_CSS#</td>
<td>Cascading Style Sheet files defined on the theme style level.</td>
</tr>
<tr>
<td>Page</td>
<td>#PAGE_TRANSITION#</td>
<td>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=&quot;page&quot;. For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;div id=&quot;#PAGE_STATIC_ID#&quot; data-role=&quot;page&quot; data-apex-page-transition=&quot;#PAGE_TRANSITION#&quot; data-apex-popup-transition=&quot;#POPUP_TRANSITION#&quot; data-url=&quot;#REQUESTED_URL#&quot;&gt;</code></td>
</tr>
<tr>
<td>Page</td>
<td>#PAGE_URL#</td>
<td>Page URL.</td>
</tr>
<tr>
<td>Page</td>
<td>#POPUP_TRANSITION#</td>
<td>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=&quot;page&quot;. For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;div id=&quot;#PAGE_STATIC_ID#&quot; data-role=&quot;page&quot; data-apex-page-transition=&quot;#PAGE_TRANSITION#&quot; data-apex-popup-transition=&quot;#POPUP_TRANSITION#&quot; data-url=&quot;#REQUESTED_URL#&quot;&gt;</code></td>
</tr>
<tr>
<td>JavaScript</td>
<td>#DIALOG_ATTRIBUTES#</td>
<td>Defines dialog attributes not declaratively supported.</td>
</tr>
<tr>
<td>JavaScript</td>
<td>#DIALOG_CSS_CLASSES#</td>
<td>Dialog CSS Classes.</td>
</tr>
</tbody>
</table>
Creating Custom Themes

### Required Dialog Page Template Attributes

To create a dialog page template, you must define the Page Template attributes described in Table 13–12.

**Table 13–12 Required Attributes for Dialog Page Templates**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name, Template Type</td>
<td>Must be defined as <strong>Dialog Page</strong>.</td>
</tr>
<tr>
<td>JavaScript, Dialog Initialization Code</td>
<td>Enter JavaScript code to initialize the page as a dialog. See field-level Help for function call syntax.</td>
</tr>
<tr>
<td>JavaScript, Dialog Closure Code</td>
<td>Enter JavaScript code to close the dialog page. See field-level Help for function call syntax.</td>
</tr>
<tr>
<td>JavaScript, Dialog Cancel Code</td>
<td>Enter JavaScript code to cancel the dialog page. See field-level Help for function call syntax.</td>
</tr>
<tr>
<td>Dialog, Width</td>
<td>Sets the default width for dialogs. The page template-level setting can be overwritten at page-level.</td>
</tr>
<tr>
<td>Dialog, Height</td>
<td>Sets the default height for dialogs. The page template-level setting can be overwritten at page-level.</td>
</tr>
<tr>
<td>Dialog, Maximum Width</td>
<td>Sets the maximum width for dialogs, in pixels. The page template-level setting can be overwritten at page-level.</td>
</tr>
<tr>
<td>Dialog, CSS Classes</td>
<td>Enter additional CSS classes you wish to be applied to dialogs. The page template-level setting can be overwritten at page-level.</td>
</tr>
<tr>
<td>Dialog, Allow Embed in Frame</td>
<td>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 <strong>Deny</strong>, checking <strong>Allow Embed In Frame</strong> for Modal Dialog will instead use <strong>Allow from same origin</strong> for dialog pages that use this template.</td>
</tr>
</tbody>
</table>

See Also: "Creating Dialog Pages" on page 7-40 and "Page Template Attributes" on page 13-42

**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.
Creating Custom Themes

Popup List of Values Template Attributes

See Also: "Creating Lists of Values at the Application Level" on page 18-18 and "Using Themes" on page 13-1

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 Use these attributes to specify how a Search field displays. Table 13–13 describes available Search Field attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Field Text</td>
<td>Defines text to display before the popup list of values search field displays.</td>
</tr>
<tr>
<td>After Field Text</td>
<td>Displays this text after displaying the search field, the search button, and the close button.</td>
</tr>
<tr>
<td>Filter Width</td>
<td>Displays the HTML INPUT TYPE = TEXT widget using this width.</td>
</tr>
<tr>
<td>Filter Max Width</td>
<td>Displays the HTML INPUT TYPE = TEXT widget using this maximum width.</td>
</tr>
<tr>
<td>Filter Text Attribute</td>
<td>Displays the HTML INPUT TYPE = TEXT widget using these attributes. This is included within the HTML input tag.</td>
</tr>
</tbody>
</table>

Buttons Use these attributes to define the button name and attributes for the Find, Close, Next, and Previous buttons.

Window Popup lists of values are executed using JavaScript. Use these attribute to control the values of scrollbars=, resizable=, width=, and height=.

Pagination Defines how row count results display.

Result Set Use these attributes to define text or HTML to display before and after a result set.

Page Attributes Use these attributes to define popup pages.

Region Templates

See Also: "Understanding Template Options" on page 13-6 and "Understanding Theme Subscriptions" on page 13-13

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 (#).

See Also: "About Regions" on page 14-4 and "Using Themes" on page 13-1
Region Template Attributes

See Also: "Understanding Template Options" on page 13-6 and "Understanding Theme Subscriptions" on page 13-13

This section describes some attributes on the Edit Region Template page. To learn more about a specific attribute, see field-level Help.

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. Application Builder accomplishes this template mapping through the assignment of a template class. Use the Translatable check box to indicate that the template contains text strings that require translation.

Definition

Region templates provide the appearance for a portion of a page called a region. Use substitution strings to indicate the existence and placement of a component within the region. #BODY# is the only required substitution string and identifies where the source of the region should be placed. All other substitution strings are optional. The following are valid substitution strings:

- #TITLE#
- #PAGE_TITLE#
- #EXPAND#
- #CHANGE#
- #BODY#
- #SUB_REGION_HEADERS#
- #SUB_REGIONS#
- #FORM_OPEN#
- #FORM_CLOSE#
- #REGION_ID#
- #REGION_STATIC_ID#
- #REGION_CSS_CLASSES#
- #REGION_ATTRIBUTES#
- #REGION_HEADER#
- #REGION_FOOTER#

When you create a button in a region position, the positions you define appear in a select list. Use the following substitution strings to define positions for the placement of buttons in a region:

- #EDIT#
- #CLOSE#
- #CREATE#
Grid Layout  Page items display within regions. If the grid type in the page template is set to Table, each item is part of an HTML table. Use this attribute to set the attributes of this table.

Sub Regions  Use Header Templates and Header Entry Templates to generate a list of region titles of all the subregions of the current region. For either template, #ENTRIES# is the only required substitution string.

Use Template to wrap a subregion with additional HTML code. #SUB_REGION# is the only required substitution string. It identifies where the source of the subregion should be placed. All other substitution strings are optional.

Substitution Strings  Lists substitution string used within templates to reference component values.

Report Templates
Report column templates provide you with control over the results of a row from a SQL query. This type of template defines a cell, not an entire row.

Each report template identifies column names using the syntax #1#, #2#, #3#, and so on. You can also name columns using column name substitution syntax such as #ENAME# or #EMPNO#. You can reference any item from your application within your template. For example, to reference an item called ABC. in your template, you could include the exact substitution string &ABC.. The actual value of ABC. would be provided by an end user editing an item in your application named ABC.

See Also:  "Using Themes" on page 13-1, "Understanding Template Options" on page 13-6, and "Understanding Theme Subscriptions" on page 13-13

Generic Column Templates and Named Column Templates
Report Column Template Attributes for Generic Column Templates
Report Column Template Attributes for Named Column Templates
About Using JavaScript in Column Templates

Generic Column Templates and Named Column Templates
Oracle Application Express includes two types of report templates:

• Generic column templates
• Named column templates
Creating Custom Themes

Generic Column Templates  A generic column template determines the appearance of a report by defining the look of the column once. This look is then repeated as many times as is necessary, based on the number of columns specified in the report’s definition. This type of template is limited to reports that have a standard row and column structure. Additional style can be applied to a report using this type of template through the use of conditions.

The following example demonstrates how to have each column use a specific style:

<td class="tabledata" align="#ALIGN#">#COLUMN_VALUE#</td>

This example assumes your page template includes a CSS containing the class tabledata. This example also demonstrates the use the substitution strings #ALIGN# and #COLUMN_VALUE#. If you actually ran this report, these substitution strings would be replaced with values generated by the results of a SQL query.

If your query uses an expression in the select list, it is a good idea to create an alias for the columns to avoid runtime errors. For example, suppose your query was as follows:

SELECT ename, (sal + comm) * 12 FROM emp

You could rewrite the query to alias the columns as follows:

SELECT ename, (sal + comm) * 12 yearly_comp FROM emp

Named Column Templates  Named column templates allow for more flexibility in report design. However, because they reference columns by name, they can only be used by reports that are based on those columns. For example:

<tr><td>#ENAME#</td><td>#SAL#</td></tr>

Although named column templates offer flexibility, you may need to create a new template for each query. You can also include a position notation. The following example demonstrates how to use following HTML and substitution strings:

<tr><td>#ENAME#</td><td>#SAL#</td></tr>
<tr><td>#1#</td><td>#2#</td></tr>

Report Column Template Attributes for Generic Column Templates

This section describes specific sections of the Edit Report Template page for Generic Column Templates. To learn more about a specific attribute, see field-level Help.

Report Template  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. Application Builder accomplishes this template mapping through the assignment of a template class. Use the Translatable check box to indicate the template contains text strings that require translation.

Before Rows  In Before Rows, enter HTML that displays once at the beginning of a report template.
Opening an HTML table is a common use of this attribute as shown in the following example:

```html
<table>
</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:

```html
<table>#TOP_PAGINATION#
</table>
```

You can also include the substitution string `#CSV_LINK#` to include support for exporting your report to comma-separated value (CSV) format, a format compatible with most spreadsheet programs.

You can also use the substitution string `#REPORT_ATTRIBUTES#`. This substitution string is set per report region on the report attributes page.

**Column Headings** Use **Before Column Heading** to display text before report heading.

Use **Column Heading Template** to colorize each column header cell. The text of this attribute must include Help to indicate where the cell heading text should be colorized. For example:

```html
<th>Item Help Text</th>
```

If you do not want any column headings, enter the following:

**OMIT**

If you do use this attribute, Application Express engine applies the default column heading template.

Use **After Column Heading** to display text after report headings.

**Before Each Row** In **Before Each Row**, enter text to display before all columns in the report. Use this attribute to open a new HTML row. Before Each Row supports the following substitution strings:

- `#ROWNUM#`
  
  Use this substitution string to specify the current row.

- `#COLCOUNT#`
  
  Use this substitution string to specify the number of columns.

- `#HIGHLIGHT_ROW#`
  
  Use this substitution string to specify the number of highlighted rows.

**Column Templates** Column templates define the look of each column. You can define up to four column templates; each can be conditional. For example, you can have different background colors for even and odd rows, or highlight rows that meet a PL/SQL defined condition.

In each Column Template, you define the look of each column. Column Templates support the substitution strings described in Table 13–14.
Consider the following example:

\[
\text{<td> #ALIGNMENT#|#COLUMN_VALUE#</td>}
\]

If you actually ran this report, these substitution strings would be replaced with values generated by the results of a SQL query.

By creating conditions, you can create a report that displays columns differently depending on whether the specified condition is met. To specify a column template be used conditionally, select a condition type from the Column Template Condition list. Valid values include:

- **Use Based on PL/SQL Expression.** Conditionally formats columns based on data in that row.

- **Use for Even Numbered Rows.** Conditionally formats even numbered rows.

- **Use for Odd Numbered Rows.** Conditionally formats odd numbered rows.

If you select **Use Based on PL/SQL Expression**, the next step is to enter a PL/SQL expression in the Column Template Expression field. For example, the following expression displays a value in bold if the value is greater than 2000:

\[
#SAL# > 2000
\]

You could also use the substitution string **#ROWNUM#.** For example:

\[
#ROWNUM# > 2000
\]

**After Each Row** In **After Each Row**, enter HTML that displays after all columns in the report display. This attribute is often used to close an HTML table row. For example:

\[
</tr>
\]

**After Rows** Use **After Rows** to specify text that should display after the last row. A common use of this attribute is to close the HTML table tag. For example:

\[
</table>
\]

The After Rows attribute supports the following substitution strings:

- **#PAGINATION#**
  
  Replaced with a pagination attribute.

- **#COLCOUNT#**
  
  Substituted at runtime with the number of columns defined in the report.

---

**Table 13-14  Column Template Substitution Strings**

<table>
<thead>
<tr>
<th>Substitution String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#ALIGNMENT#</td>
<td>Determines the column alignment. Specified by the user.</td>
</tr>
<tr>
<td>#COLCOUNT#</td>
<td>Defines the count of the number of columns.</td>
</tr>
<tr>
<td>#COLNUM#</td>
<td>Defines the current column number.</td>
</tr>
<tr>
<td>#COLUMN_HEADER#</td>
<td>Defines the column header.</td>
</tr>
<tr>
<td>#COLUMN_VALUE#</td>
<td>Replaced with the value of the column.</td>
</tr>
<tr>
<td>#ROWNUM#</td>
<td>Specifies the current row number.</td>
</tr>
</tbody>
</table>
- **#CSV_LINK#**
  Substituted at runtime with CSV download link if CSV export is enabled. Otherwise the string is not shown.

- **#EXTERNAL_LINK#**
  Substituted at runtime with the download link for exports using external processing engine for (for example, PDF export). If this feature is not enabled, the string is not shown.

**Row Highlighting** *Background color for checked row* controls the background color of a report row when the row selector is checked. *Background color for current row* controls the background color of a report row when the user moves the mouse over the row.

**Pagination Subtemplate** The Pagination Subtemplate section contains attributes for editing the Pagination Template, Next Page Template, Previous Page Template, Next Set Template, and Previous Template. Pagination Subtemplates support the substitution strings #PAGINATION_NEXT#, #PAGINATION_NEXT_SET#, #PAGINATION_PREVIOUS# and #PAGINATION_PREVIOUS_SET#. Table 13–17 describes these templates.

<table>
<thead>
<tr>
<th>Pagination Subtemplate Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| Pagination Template              | Applies to the entire pagination subtemplate. For example:  
  <span class="instructiontext">#TEXT#</span>  
  Use the substitution string #TEXT# to specify where you want the pagination subtemplate to display.  
  Use the other Pagination Subtemplate attributes to modify individual items.  
| Next Page Template                | Enter HTML to modify how the Next Page portion of the pagination subtemplate appears. For example:  
  <a href="#LINK#">next</a>  
| Previous Page Template            | Enter HTML to modify how the Previous Page portion of the pagination subtemplate appears. For example:  
  <a href="#LINK#">previous</a>  
| Next Set Template                 | Enter HTML to modify how the Next Set portion of the pagination subtemplate appears. For example:  
  <a href="#LINK#">next set</a>  
| Previous Set Template             | Enter HTML to modify how the Previous Set portion of the pagination subtemplate appears. For example:  
  <a href="#LINK#">previous set</a>  

**Comments** Use this attribute to record comments about this component.
Creating Custom Themes

**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**  
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. Application Builder accomplishes this template mapping through the assignment of a template class.

**Subscription** Use Subscription to apply an existing template to the current application. When you select an existing template, you become a subscriber to that template.

To load a new copy of a master template, click **Refresh**.

**Row Templates**  
Row templates define the look of each column. You can define up to four row templates, each of which can be conditional.

In each Row Template, you define the look of each row. Row Templates support the substitution strings described in Table 13–16.

<table>
<thead>
<tr>
<th>Table 13–16</th>
<th>Row Template Substitution Strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitution String</td>
<td>Description</td>
</tr>
<tr>
<td>#ALIGNMENT#</td>
<td>Determines the row alignment. Specified by the user.</td>
</tr>
<tr>
<td>#COLCOUNT#</td>
<td>Defines the count of the number of columns.</td>
</tr>
<tr>
<td>#COLNUM#</td>
<td>Defines the current column number.</td>
</tr>
<tr>
<td>#COLUMN_HEADER#</td>
<td>Defines the column header.</td>
</tr>
<tr>
<td>#COLUMN_VALUE#</td>
<td>Replaced with the value of the column.</td>
</tr>
<tr>
<td>#ROWNUM#</td>
<td>Specifies the current row number.</td>
</tr>
</tbody>
</table>

By creating conditions, you can create a report that displays rows differently depending on whether the specified condition is met. To specify a row template be used conditionally, select a condition type from the Column Template Condition list. Valid values include:

- **Use Based on PL/SQL Expression.** Conditionally formats columns based on data in that row.
- **Use for Even Numbered Rows.** Conditionally formats even numbered rows.
- **Use for Odd Numbered Rows.** Conditionally formats odd numbered rows.

If you select **Use Based on PL/SQL Expression**, the next step is to enter a PL/SQL expression in the Column Template Expression field. For example, the following expression displays a value in bold if the value is greater than 2000:

`#SAL# > 2000`
You could also use the substitution string \#ROWNUM\#. For example:

\#ROWNUM\# > 2000

**Column Headings** Use this template to add color to each column header cell. The text of this attribute must include help to indicate where the cell heading text should be colorized. If you do not enter a Column Heading Template, then a default column header template is applied. If you do not want any column headings, then enter OMIT.

For example:

```html
<th #ALIGNMENT#>#COLUMN_HEADER#</th>
```

**Before first and after last row text** In *Before Rows*, enter HTML that displays once at the beginning of a report template. Opening an HTML table is a common use of this attribute, as shown in the following example:

```html
<table>
</table>
```

You can identify column headers using the syntax \#1\#, \#2\#, \#3\#. For example:

```html
<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:

```html
<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:

```html
</table>
```

The After Rows attribute supports the following substitution strings:

- \#PAGINATION\#
  - Replaced with a pagination attribute.
- \#COLCOUNT\#
  - Substituted at runtime with the number of columns defined in the report.

**Pagination** The Pagination section contains attributes for editing the Pagination Template, Next Page Template, Previous Page Template, Next Set Template, and Previous Template. Pagination Subtemplates support the substitution strings \#PAGINATION_NEXT\#, \#PAGINATION_NEXT_SET\#, \#PAGINATION_PREVIOUS\# and \#PAGINATION_PREVIOUS_SET\#. Table 13–17 describes these templates.
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

### Table 13–17 Pagination Subtemplate Attributes

<table>
<thead>
<tr>
<th>Pagination Subtemplate Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pagination Template</td>
<td>Applies to the entire pagination subtemplate. For example:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;span class=&quot;instructiontext&quot;&gt;#{TEXT}&lt;/span&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Use the substitution string <code>#TEXT#</code> to specify where you want the pagination subtemplate to display.</td>
</tr>
<tr>
<td></td>
<td>Use the other Pagination Subtemplate attributes to modify individual items.</td>
</tr>
<tr>
<td>Next Page Template</td>
<td>Enter HTML to modify how the Next Page portion of the pagination subtemplate appears. For example:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;a href=&quot;#LINK&quot;&gt;next&lt;/a&gt;</code></td>
</tr>
<tr>
<td>Previous Page Template</td>
<td>Enter HTML to modify how the Previous Page portion of the pagination subtemplate appears. For example:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;a href=&quot;#LINK&quot;&gt;previous&lt;/a&gt;</code></td>
</tr>
<tr>
<td>Next Set Template</td>
<td>Enter HTML to modify how the Next Set portion of the pagination subtemplate appears. For example:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;a href=&quot;#LINK&quot;&gt;next set&lt;/a&gt;</code></td>
</tr>
<tr>
<td>Previous Set Template</td>
<td>Enter HTML to modify how the Previous Set portion of the pagination subtemplate appears. For example:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;a href=&quot;#LINK&quot;&gt;previous set&lt;/a&gt;</code></td>
</tr>
</tbody>
</table>

**Comments** Use this attribute to record comments about this component.

### About Using JavaScript in Column Templates

You can conditionally display HTML depending upon values in the database using JavaScript. The following example displays an HTML row only if the `GROUP_DESC` query column is not null:

```javascript
<script language="javascript">
IF ( "#GROUP_DESC#" != "" )
document.writeln("<TR>
<TD BGCOLOR="#336699">;</TD>
</TR>
<TD>#GROUP_DESC#</TD>
</TR>
</script>
```

**See Also:** "About Regions" on page 14-4
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.

- **About Uploading Cascading Style Sheets**
- **Referencing an Uploaded Cascading Style Sheet in the Page Template**

**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" on page 13-3 and "Using Theme Roller" on page 13-4.

### 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" on page 18-8 and "Managing Static Application Files" on page 18-8.

**See Also:** "Referencing an Uploaded Cascading Style Sheet in the Page Template" on page 13-59

### 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 **Application Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Themes**.
   
   The Themes page appears.
5. On the Tasks list, click **View Templates**.
6. Select the name of the page template you want to edit.
7. 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
<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
<html>
<head>
    <title>#TITLE#</title>
```
Using Custom Cascading Style Sheets

#HEAD#
<link rel="stylesheet" href="#WORKSPACE_IMAGES#sample3.css" type="text/css">
</head>

This section describes the ways you can customize your 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.

- About Page Layout in Oracle Application Express
- Creating a Global Page to Display Components on Every Page
- About Regions
- Managing Regions in Page Designer
- Managing Regions in Component View
- About Incorporating Content from Other Web Sites
- About Managing Images
- Rendering HTML Using Custom PL/SQL

See Also: "Adding Navigation" on page 16-1, "Managing the Application User Interface" on page 7-29, and "Using Themes and Theme Styles" on page 13-1

### 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.
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

To select a Printer Friendly template:

1. Navigate to the Themes page:
   a. On the Workspace home page, click Application Builder.
   b. Select an application.
   c. Click Shared Components.
   d. Under User Interface, select Themes.
      The Themes page appears.
2. Change to a report view by clicking the View Report icon.
3. In the Themes list, click the theme name.
   The Create/Edit Theme page appears.
4. Scroll down to Component Defaults.
5. Make a selection from the Printer Friendly Page list.
6. Click Apply Changes.

See Also: "Editing Themes" on page 13-8 and "Creating a Theme" on page 13-15

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.
Creating a Global Page to Display Components on Every Page

This section explains how to create a Global page. A global page (previously referred to as Page 0) functions as a master page. You 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.

See Also:  "Managing Pages in a Database Application" on page 7-36, "Understanding Conditional Rendering and Processing" on page 2-12 and "Available Conditions" on page C-1

Creating a Global Page

To create a Global page:

1. On the Workspace home page, click Application Builder.
2. Select an application.
   The Application home page appears.
3. Click the Create Page button.
4. On Create Page:
   a. User Interface - Displays the selected user interface for the application.
   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" on page 7-29

Navigating to the Global Page

To navigate to the Global page:

1. On the Workspace home page, click the Application Builder icon.
   The Application Builder home page appears.
2. Select an application.
   The Application home page appears.
3. Select Global Page (or Page Zero is this is an older application).
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.


Managing Regions in Page Designer

Each page in an Oracle Application Express application contains one or more regions. Regions function as containers for content. This section explains how to create and manage regions.

- Creating a Region in Page Designer
- Managing Regions in Page Designer
- Copying a Region in Page Designer
- Deleting Regions in Page Designer

See Also: "About Regions" on page 14-4 and "Managing Regions in Component View" on page 14-10

Creating a Region in Page Designer

To create a region in Page Designer:


   Page Designer appears. 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. From the Gallery select a region and drag it to the appropriate location in Grid Layout.

   Based on the region type, 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.

   See Also: "About Regions" on page 14-4 and "Managing Regions in Component View" on page 14-10
Managing Regions in Page Designer

- **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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

4. Edit the appropriate attributes in the Property Editor.
   Required attributes display a red triangle in the upper left corner next above attribute label.

   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Click **Save**.

Managing Regions in Page Designer

A region is an area of a page that serves as a container for content. This section explains how to alter the appearance of a page by editing region attributes.

- **Editing Region Attributes in Page Designer**
- **Controlling Region Positioning**
- **Specifying a Region Header and Footer**
- **Enabling Users to Customize a Page**
- **Utilizing Region Caching in Page Designer**
- **Specifying a Static Region ID**
- **Adding a Region Image**
- **Creating a Region Display Selector**

**Editing Region Attributes in Page Designer**

To edit region attributes:

   Page Designer appears.

2. Select the region in the Rendering tree or Grid Layout.
   The Property Editor displays attributes for the region.

3. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

4. Edit the appropriate attributes in the Property Editor.
   To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

5. Click **Save**.
Controlling Region Positioning
When you create a region, you can specify its position by editing the Layout, Position attribute in the Property Editor. The template positions defined within the current theme for the page’s user interface determine the available selections.

Specifying a Region Header and Footer
In addition to the body content of a region, you can specify additional HTML to be placed above and below a region by editing the Header and Footer attributes in the Property Editor.

For all report regions, the substitution string #TIMING# shows the elapsed time in seconds used when rendering a region. You can use this substitution string for debugging purposes.

For classic report regions, the region footer supports the following substitution strings:

- #ROWS_FETCHED# shows the number of rows fetched by the Oracle Application Express reporting engine (the page size). You can use these substitution strings to display customized messages to the user. For example:

  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#

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” on page 13-23

To enable end-user customization:


   Page Designer appears.

2. Select the region in the Rendering tree or Grid Layout.

   The Property Editor displays attributes for the region.

3. Click the Collapse All icon and then select Show All from the Property Editor Menu.

4. Expand Customization. From Customizable, select one of the following:
Managing Regions in Page Designer

Controlling Page Layout

1. Not Customizable By End Users
2. Customizable and Not Shown By Default
3. Customizable and Shown By Default

5. Click Save.

Utilizing Region Caching in Page Designer

This section describes the advantages of using region caching to improve the performance of static regions.

Tip: Region caching is not available for interactive report regions.

About Region Caching

Enabling Region Caching in Page Designer

See Also: "Managing Page and Region Caching" in Oracle Application Express Administration Guide

About Region Caching

Enabling region caching is an effective way to improve the performance of static regions such as regions containing lists that do not use conditions or regions containing static HTML.

When you enable region caching, the Application Express engine renders a region from a cached (or stored) repository instead of rendering it dynamically. Keep in mind that the actual session identifiers are not cached. Instead, the Application Express engine caches a &SESSION. substitution string and the current session rendering the cached region is substituted on display. For example, if a region contains a link and the link includes a session, the exact session is not cached to ensure that the links works for all sessions.

The Application Express engine only renders a region from cache if it meets the defined condition. Additionally, regions can be cached specific to a user or cached independent of a user.

See Also: "Server Cache" on page 9-20

Enabling Region Caching in Page Designer

To enable region caching:


Page Designer appears.

2. Select the region in the Rendering tree or Grid Layout.

The Property Editor displays attributes for the region.

3. Click the Collapse All icon and then select Show All from the Property Editor Menu.

4. Expand Server Cache. From 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.
■ **Cache Timeout** - Displays if you select you select **Enabled** or **Cached by User**. Select how long a cached page is valid. Subsequent page requests do not utilize the cached page if the timeout expires.

■ **Type** - 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.

5. Click **Save**.

**Specifying a Static Region ID**

Specifying a static region ID is useful when creating custom JavaScript or cascading style sheets. You can use the Static ID attribute on the Edit Region page to uniquely identify a region. You can then reference the region using the `#REGION_STATIC_ID#` substitution string in a region templates, the header, the footer, or the body.

A static region ID is included by assigning it as an HTML element ID to the region container object (that is, the table or DIV). The ID of an HTML element must be unique for the entire page. For example, to keep the page HTML valid you cannot have a button and region with the same ID.

To specify a static region ID:


   Page Designer appears.

2. Select the region in the Rendering tree or Grid Layout.

   The Property Editor displays attributes for the region.

3. Click the **Collapse All** icon and then select **Show All** from the Property Editor Menu.

4. Expand Advanced and locate Static ID.

5. 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.

6. Click **Save**.

**Adding a Region Image**

Use the Region Image and Image tag attributes to add an image to the upper left side of a region.

To add a region image:


   Page Designer appears.

2. Select the region in the Rendering tree or Grid Layout.

   The Property Editor displays attributes for the region.
3. Click the **Collapse All** icon and then select **Show All** from the Property Editor Menu.

4. Expand **Advanced** and specify the following:
   - Region Image - Enter a reference to an image that displays in the upper left of the region. There is no control over the table tag used to display this image.
   - Image Tag Attributes - Enter image tag attributes included in the image HTML. This can be used for attributes such as height, width or image ALT text (if the image is non-decorative).
     If no ALT is defined in the image tag attributes, Application Express defaults to rendering the image as a decorative image, for example with an empty ALT tag.

5. Click **Save**.

**Creating a Region Display Selector**

Region Display Selector region enables you to include show and hide controls for each region on a page.

To create a Region Display Selector:

1. Create the page and regions to be included in the Region Page Selector.

2. For each region to be included in the selector, select **Yes** for the Region Display Selector attribute. See “Managing Regions in Page Designer” on page 14-5.

3. Create a region of type Region Display Selector.
   - From the Gallery select the region type **Region Display Selector** and drag it to the appropriate location in Grid Layout.
   - Edit the region attributes in the Property Editor:
     - Click the **Collapse All** icon and then select **Show All** from the Property Editor Menu.
     - Expand **Identification** and edit the Title.
     - Expand **Advanced** and select **Yes** for Region Display Selector.
   - Click **Save**.

**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:

   Page Designer appears.

2. Right-click the region in Grid Layout and select Copy To and select a destination from the submenu.
Tip: To copy a region to another page, select the region in the Rendering tree, right-click, and select Copy to Other Page. Follow the on-screen instructions.

Deleting Regions in Page Designer

To delete regions:


Page Designer appears.

2. Select one or more regions in the Rendering tree.

3. Right-click and select Delete.

4. Click Save.

Managing Regions in Component View

Each page in an Oracle Application Express application contains one or more regions. Regions function as containers for content. This section explains how to create and manage regions.

- About Region Types When Running the Create Region Wizard
- Creating a Region in Component View
- Managing Regions in Component View
- Creating a Region Display Selector
- Copying a Region in Component View
- Deleting Regions in Component View

See Also: "About Regions" on page 14-4 and "Managing Regions in Page Designer" on page 14-4

About Region Types When Running the Create Region Wizard

When you create a region, you select a region type. The Application Express engine interprets a region differently based on the region type you select. Table 14–1, "Region Types When Running the Create Region Wizard" describes the available region types. The available region types differ depending the selected user interface (for example, Desktop and jQuery Mobile Smartphone).
<table>
<thead>
<tr>
<th>Region Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Content</td>
<td>Functions as containers for items and contains the HTML you provide. Any HTML you type may contain substitution strings.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Multiple Static Content</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Report, Interactive Report</td>
<td>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.</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> jQuery Mobile Smartphone does not support interactive reports.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> “Customizing Interactive Reports as an End User” on page 10-57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report, Classic Report</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> “Developing Reports” on page 10-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> “Implementing Web Services” on page 25-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report, List View</td>
<td>List view is optimized to display data and provide easy navigation on Smartphones.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Desktop applications do not support list view.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report, Column Toggle Report</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Desktop applications do not support this report format.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region Type</td>
<td>Description</td>
<td>Desktop</td>
<td>jQuery Mobile</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>Reflow Report</td>
<td>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. <strong>Note:</strong> Desktop applications do not support this report format.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Form, Form on a Procedure</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Form, Form on a Table or View</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Form, Form on a Table with List View</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
</tr>
<tr>
<td>Form, Form on a Table with Report</td>
<td>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. <strong>Tip:</strong> This option does not support tables having more than 127 columns. Selecting more than 127 columns generates an error.</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Form, Master Detail Form</td>
<td>Creates a form that displays a master row and multiple detail rows within a single HTML form. Users can query, insert, update, and delete values from two tables or views. You choose the tables on which to build the master report and detail form. You choose the tables on which to build the master report and detail form. <strong>See Also:</strong> &quot;Managing Master Detail Forms&quot; on page 11-12</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Form, Tabular Form</td>
<td>Creates a form to perform update, insert, and delete operations on multiple rows in a database table. You choose the table and columns on which to build the form. <strong>See Also:</strong> &quot;Managing Tabular Forms&quot; on page 11-6</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Form, Form on a SQL Query</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>Form, Summary Page</td>
<td>Creates a read-only version of a form. Use this option to create a confirmation page at the end of a wizard.</td>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>
### Table 14–1  (Cont.) Region Types When Running the Create Region Wizard

<table>
<thead>
<tr>
<th>Region Type</th>
<th>Description</th>
<th>Desktop</th>
<th>jQuery Mobile</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form, Form on Web Service</td>
<td>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.</td>
<td>Not available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating a Form on a Web Service&quot; on page 25-17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form, Form and Report on Web Service</td>
<td>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.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating an Input Form and Report on a Web Service&quot; on page 25-15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plug-ins</td>
<td>Plug-ins allow developers to declaratively extend the built-in types available with Application Express.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Implementing Plug-ins&quot; on page 19-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chart</td>
<td>Create line, bar, or pie charts based on SQL queries.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating Charts&quot; on page 12-22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Map Chart</td>
<td>Create declaratively defined Flash maps.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating Maps&quot; on page 12-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tree</td>
<td>Create hierarchical navigational control (or tree) based on a SQL query executed at runtime.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating Trees&quot; on page 15-49</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calendar</td>
<td>Create a region to contain a calendar with monthly, weekly, daily, and list views.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating Calendars&quot; on page 12-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>List</td>
<td>Create a list region containing a shared collection of links called list.</td>
<td>Available</td>
<td>Not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating Lists&quot; on page 18-20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breadcrumb</td>
<td>Create a region to contain a breadcrumb. Breadcrumbs are a hierarchical list of links.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating Breadcrumbs&quot; on page 16-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PL/SQL Dynamic Content</td>
<td>Create a regions based on PL/SQL that enable you to render any HTML or text using the PL/SQL Web Toolkit.</td>
<td>Available</td>
<td>Available</td>
<td></td>
</tr>
</tbody>
</table>
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 Component View" on page 9-5.
2. Under Regions, click the Create icon.
   The Create Region Wizard appears.
3. Select a region type. See Table 14–1, "Region Types When Running the Create Region Wizard".
   The Create Region Wizard appears.
4. Follow the on-screen instructions.

Managing Regions in Component View

A region is an area of a page that serves as a container for content. This section explains how to alter the appearance of a page by editing region attributes.

- Editing Region Attributes in Component View
- How Region Attributes Affect Page Layout
- Specifying a Region Header and Footer
- Enabling Users to Customize a Page
- Utilizing Region Caching in Component View
- Specifying a Static Region ID
- Adding a Region Image
- Editing Multiple Regions at Once in Component View
- Managing Subregions in Component View
Editing Region Attributes in Component View

To edit region attributes in Component View:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
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" on page 19-8.

3. Edit the appropriate attributes.

   To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
4. Click **Apply Changes**.

   **See Also:** "How Region Attributes Affect Page Layout" on page 14-15, "Controlling Region Positioning" on page 14-6, and "Specifying a Region Header and Footer" on page 14-17

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
- Controlling Region Positioning

**Component View Region Attributes Affecting Page Layout** Table 14–2, "Region Attributes Affecting Page Layout" describes region attributes that affect the layout of a page.

<table>
<thead>
<tr>
<th>Region, Attribute</th>
<th>Description</th>
<th>See Also:</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface, Sequence</td>
<td>Specifies the display order of the regions within the page.</td>
<td>&quot;Managing Subregions in Component View&quot; on page 14-21</td>
</tr>
<tr>
<td>User Interface, Parent Regions</td>
<td>Defines the parent region to which the current region belongs. By selecting a Parent region, you create a parent/child relationship between two regions.</td>
<td>&quot;Managing Subregions in Component View&quot; on page 14-21</td>
</tr>
<tr>
<td>User Interface, Display Point</td>
<td>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.</td>
<td>&quot;Managing Subregions in Component View&quot; on page 14-21</td>
</tr>
<tr>
<td>User Interface, Template</td>
<td>Determines the look of the region. To view template attributes, click the template name on the page.</td>
<td>&quot;Creating Custom Themes&quot; on page 13-23 and &quot;Region Templates&quot; on page 13-49</td>
</tr>
<tr>
<td>User Interface, Item Display Position</td>
<td>Determines if page items are displayed above or below the main region content.</td>
<td></td>
</tr>
</tbody>
</table>
Managing Regions in Component View

Table 14–2 (Cont.) Region Attributes Affecting Page Layout

<table>
<thead>
<tr>
<th>Region, Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Layout, Start New Grid</td>
<td>Determines whether this region displays in a new grid or in the same grid as the previous region.</td>
</tr>
<tr>
<td>Grid Layout, Start New Row</td>
<td>Determines whether this region displays on the same row as the previous region or whether it displays on the next row.</td>
</tr>
<tr>
<td>Grid Layout, Column</td>
<td>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.</td>
</tr>
<tr>
<td>Grid Layout, Column Span</td>
<td>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.</td>
</tr>
<tr>
<td>Grid Layout, Column CSS Classes</td>
<td>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.</td>
</tr>
<tr>
<td>Grid Layout, Column Attributes</td>
<td>Enter additional attributes which are set for the grid column. This value replaces the #ATTRIBUTES# substitution placeholder in the grid layout column template.</td>
</tr>
<tr>
<td>Attributes, Region CSS Classes</td>
<td>Enter additional CSS classes as space delimited list which are substituted if the region template contains the substitution string #REGION_CSS_CLASSES#.</td>
</tr>
<tr>
<td>Attributes, Region</td>
<td>Region templates that contain the substitution string #REGION_ATTRIBUTES# are substituted with this value.</td>
</tr>
<tr>
<td>Attributes, Region Display Selector</td>
<td>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.</td>
</tr>
<tr>
<td>Attributes, Region Image</td>
<td>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.</td>
</tr>
<tr>
<td>Attributes, Image Tag Attributes</td>
<td>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).</td>
</tr>
<tr>
<td>Header and Footer</td>
<td>Specifies HTML text to be displayed at the top of the region (just before the #BODY# content).</td>
</tr>
</tbody>
</table>
Managing Regions in Component View

Controlling Page Layout

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.

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:
  
  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:
  
  Rows #FIRST_ROW_Fetched# through #LAST_ROW_Fetched# of #ROWS_FETCHED# displayed

**See Also:** "About Region Types When Running the Create Region Wizard" on page 14-10

<table>
<thead>
<tr>
<th>Region, Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditions</td>
<td>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: &quot;Understanding Conditional Rendering and Processing&quot; on page 2-12 and &quot;Optimizing a Page for Printing&quot; on page 14-2</td>
</tr>
<tr>
<td>Read Only, Read Only Condition Type</td>
<td>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.</td>
</tr>
<tr>
<td>Customization</td>
<td>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: &quot;Enabling Users to Customize a Page&quot; on page 14-18</td>
</tr>
</tbody>
</table>
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" on page 13-23

To enable end-user customization:
1. Navigate to the page. See "Viewing a Page in Component View" on page 9-5.
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.

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
■ Enabling Region Caching
■ Managing Cached Regions

See Also: "Managing Page and Region Caching" in Oracle Application Express Administration Guide

About Region Caching
Enabling region caching is an effective way to improve the performance of static regions such as regions containing lists that do not use conditions, or regions containing static HTML.

When you enable region caching, the Application Express engine renders a region from a cached (or stored) repository instead of rendering it dynamically. Keep in mind that the actual session identifiers are not cached. Instead, the Application Express engine caches a &SESSION. substitution string and the current session rendering the cached region is substituted on display. For example, if a region contains a link and the link includes a session, the exact session is not cached to ensure that the links works for all sessions.
The Application Express engine only renders a region from cache if it meets the defined condition. Additionally, regions can be cached specific to a user or cached independent of a user.

**See Also:**  "Server Cache" on page 9-20 and "Managing Cached Regions" on page 14-19

**Enabling Region Caching** To enable region caching:

1. Navigate to the page.  
   See “Viewing a Page in Component View” on page 9-5.

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**.

**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 Component View" on page 9-5.

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*
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 Component View” on page 9-5.
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.

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 Component View” on page 9-5.
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.

Editing Multiple Regions at Once in Component View
This section describes ways to edit multiple regions at once.

■ Editing Multiple Attributes Across All Regions
■ Accessing Region Utilities

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.
To edit multiple attributes across all regions:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
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**.

**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 Component View" on page 9-5.
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.

**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**
- **Creating a Subregion in Component View**
- **Creating a Subregion by Editing Region Attributes**

**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" on page 13-49

**Creating a Subregion in Component View** To create a subregion:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
2. Under Regions, click the **Create** icon.

3. Select a region type. See Table 14-1, "Region Types When Running the Create Region Wizard".

   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.

**Creating a Subregion by Editing Region Attributes**

To create a subregion by editing region attributes:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

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” on page 1-21.

6. Click **Apply Changes**.

**See Also:** "How Region Attributes Affect Page Layout" on page 14-15, "Controlling Region Positioning" on page 14-6, and "Specifying a Region Header and Footer" on page 14-17

**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:
Managing Regions in Component View

Controlling Page Layout

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

2. Under Regions, select the region name.
   The Edit Region page appears.

3. Under Attributes, change Region Display Selector to Yes.

   d. Click Apply Changes.

3. Create a region of type Region Display Selector:


   b. Under Regions, click the Create icon.
      The Create Region Wizard appears.

   c. Select the region type Region Display Selector.

   d. Click Create.

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 Component View" on page 9-5.

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.

Deleting Regions in Component View

This section describes how to delete regions.

- Deleting a Region
- Deleting Multiple Regions at Once

Deleting a Region

To delete a region:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
2. Under Regions, select the region name.
   The Edit Region page appears.
3. Click Delete.

Deleting Multiple Regions at Once
To delete multiple regions at once:
1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
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.

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)
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" on page 14-5 and "Editing Region Attributes in Component View" on page 14-15

**Working with SSL-Enabled URLs**

If you call a SSL-enabled URL (for example, by invoking a Web service), or create a region of type URL that is SSL-enabled, you must create a wallet. A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL.

**Tip:** See "Configuring Wallet Information" in *Oracle Application Express Administration Guide*

**About Managing Images**

To reference an image within an application, you must upload it to the static file repository. During the upload process, you specify whether the file is available to all applications or just a specific application.

**See Also:** "Managing Static Application Files" on page 18-8 and "Managing Static Workspace Files" on page 18-12

**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.

**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 the Application Attributes" on page 7-14.
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:

```
htp.p(TO_CHAR(SYSDATE,'Day Month DD, YYYY'));
```

This next example accesses tables:

```sql
DECLARE
    l_max_sal NUMBER;
BEGIN
    SELECT max(sal) INTO l_max_sal FROM emp;
    htp.p('The maximum salary is: '||TO_CHAR(l_max_sal,'999,999.00'));
END;
```

See Also:

- Oracle Database Development Guide for information about developing web applications with PL/SQL.
- Oracle Database PL/SQL Packages and Types Reference for information about htp packages.
Managing Database Application Controls

This section explains how to create and manage database application page controls (including page-level items, dynamic actions, buttons, and trees) and how to incorporate JavaScript.

- Understanding Page-Level Items
- Managing Page-Level Items in Page Designer
- Managing Page-Level Items in Component View
- Managing Dynamic Actions
- Creating Buttons
- Creating Trees
- About Incorporating JavaScript into an Application

See Also: "Developing Reports" on page 10-1, "Controlling Page Layout" on page 14-1, "Adding Navigation" on page 16-1, and "Manually Refreshing Oracle Application Express Components" on page 19-15

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
- About Item Naming Conventions
- Referencing Item Values
- About Referencing Items Using JavaScript
- Working with Multiple Select List Item

See Also: "Managing Page-Level Items in Page Designer" on page 15-7 and "Managing Page-Level Items in Component View" on page 15-16

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" on page 18-15

### 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.
- Cannot contain letters outside the base ASCII character set.

As a best practice Oracle recommends including the page number when naming items. By default, wizards prefix page item names with `P<page no>_<item name>` (for example, `P1_NAME`).

### Referencing Item Values

You can reference item values stored in session state in regions, computations, processes, validation, and branches. Table 15–1 describes the supported syntax for referencing item values.

**See Also:** "Managing Session State Values" on page 2-20

<table>
<thead>
<tr>
<th>Type</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL</td>
<td>:MY_ITEM</td>
<td>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.</td>
</tr>
</tbody>
</table>
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:

  ```plsql
  BEGIN
  :MY_ITEM := 'new value';
  END;
  ```

- Pass the value on a URL reference using `f?p` syntax. For example:

  ```url
  f?p=100:101:10636547268728380919::NO::MY_ITEM:ABC
  ```

- Set the value using a computation. Computations are designed to set item values. For example:

  ```plsql
  TO_CHAR(SYSDATE,'Day DD Month, YYYY');
  ```

- Use the PL/SQL API to set an item value within a PL/SQL context. For example:

  ```plsql
  APEX_UTIL.SET_SESSION_STATE('MY_ITEM',SYSDATE);
  ```

  **See Also:** "Clearing Session State" on page 2-22, Oracle Application Express API Reference, and "Understanding Cross-Site Scripting Protection" on page 20-16

**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,
Understanding Page-Level Items

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 `$v('P1_FIRST_NAME')` and `$s('P1_FIRST_NAME', 'Joe')`. Consider the following example:

```javascript
function showFirstName()
    alert('First Name is ' + $v('P1_FIRST_NAME'))
}; function setFirstName(pFirstName)
    $s('P1_FIRST_NAME', pFirstName);
};
```

These functions can be called by other JavaScript functions or with the Execute JavaScript code dynamic action.

See Also: "APEX_JAVASCRIPT" in Oracle Application Express API Reference

Working with Multiple Select List Item

This section describes how to handle values returned from multiple select list item.

- About Handling Values Returned from a Multiple Select List Item
- Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values
- Creating a Shuttle Item

About Handling Values Returned from a Multiple Select List Item

A multiple select item renders as a multiple select list form element which can be either a Multiselect List or Shuttle item type. When submitted, selected values are returned in a single colon-delimited string. You can handle values in this format in three ways:

- Using the `INSTR` function
- Using the `APEX_UTIL.STRING_TO_TABLE` function
- Creating a shuttle

Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values

Suppose you had a report on the `EMP` and `DEPT` tables that is limited by the departments selected from a Department multiple select list. First, you create the multiple select item, `P1_DEPTNO`, using the following query:

```sql
SELECT dname, deptno
FROM dept
```

Second, you return only those employees within the selected departments as follows:

```sql
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

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
- Creating a Shuttle Item on the Form Page in Page Designer
- Creating a Shuttle Item on the Form Page in Component View

Creating a Report and From on a Table
To create report and form on a table:
1. On the Workspace home page, click the Application 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 Form.
   c. Click Next.
5. Select Form on a Table with Report and click Next.
6. On Report Page Attributes:
   a. Edit the attributes as needed.
   b. Click Next.
7. On Data Source:
Understanding Page-Level Items

- Table/View Owner - Select the owner of the table on which you are building a report.
- Table/View Name - Select DEPT.
- Click Next.

8. Accept the remaining defaults and follow the on-screen instructions.

To learn more about an option, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

Creating a Shuttle Item on the Form Page in Page Designer

To create a shuttle on the form page:

1. Go to the form page and view it in Page Designer. See "Viewing Field-Level Help" on page 1-21.
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:
   - Name - Enter PX_EMP_LIST.
   - Type - Select Shuttle.

4. Under List of Values:
   - Type - Select SQL Query.
   - SQL Statement - Enter:
     ```sql
     SELECT ename, empno FROM emp ORDER BY 1
     ```

5. Under Source:
   - Type - Select SQL Query (return colon separated value).
   - SQL Query - Enter:
     ```sql
     SELECT empno FROM emp WHERE deptno = :P2_DEPTNO ORDER BY ename
     ```

6. Under Source:
   - Type - Select SQL Query (return colon separated value)
   - Item Source Value - Enter:
     ```sql
     SELECT empno FROM emp WHERE deptno = :P8_DEPTNO ORDER BY ename
     ```

7. Click Save.
8. Click Save and Run to view the page.

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 Component View" on page 9-5.
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:
     
     ```sql
     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:
     
     ```sql
     SELECT empno FROM emp WHERE deptno = :P8\_DEPTNO ORDER BY ename
     ```
   - Click Create Item.
9. Click Run to view the page. You shuttle item should look similar to the following illustration.

---

**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
- Creating Page-Level Items
- Editing Page-Level Items in Page Designer

**See Also:** "Understanding Page-Level Items" on page 15-1 and "Managing Page-Level Items in Component View" on page 15-16

---

**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:

2. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.
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. You shuttle item should look similar to the following illustration.

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 Tree
- Adding a Page-Level Item from the Gallery
- Creating a Static List of Values
- Creating a Cascading List of Values

See Also: "About the Differences Between Page Items and Application Items" on page 15-1

Creating a Page-Level Item from the Rendering Tree

To create a page-level item from the Rendering tree:

1. View the page to contain the item in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.
2. If necessary, create a region to contain the item. See "About Regions" on page 14-4.
3. In the Rendering tree, right-click the region to contain the item and select Create Page Item.

The Property Editor - Page Items appears.

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. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.
5. Edit the appropriate attributes in the Property Editor.

Tip: To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane.

6. Click Save.

Adding a Page-Level Item from the Gallery

To create a page-level item by adding if from the Gallery:

1. View the page to contain the item in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.
2. If necessary, create a region to contain the item. See "About Regions" on page 14-4.
3. In the Gallery, click the Items tab.

    Passing the cursor over an item displays a tooltip that describes it.
4. Select an item and drag it to the appropriate location in Grid Layout.

   **Tip:** You also right-click the item, click **Add To**, and select the appropriate location.

The Property Editor - Page Items appears.

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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

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**.

### 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" on page 18-18

To create a static list of values:


3. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

4. Edit the appropriate attributes in the Property Editor.

   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Under Identification:
   a. Name - Enter a name. See "About Item Naming Conventions" on page 15-2.
   b. Type - Select **Select List**.

6. Under List of Values:
   a. Type - 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:
Managing Page-Level Items in Page Designer

- The first keyword may be STATIC or STATIC2.
  - If the text starts with STATIC:, the list values display alphabetically. Alternatively, if the text starts with STATIC2:, the list values display 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.

7. Click Save.

8. Click Save and Run to view the page.

The examples that follow demonstrate syntax for three different static LOVs.

**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
```

**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
```

**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
```

**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. Application 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:
    
    ```sql
    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:
    
    ```sql
    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.

**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
- Defining Default Values
- Configuring Page Item Security
- Creating a Quick Pick Selection
- Displaying Conditional Page Items
- Displaying Read Only Page Items
- Viewing Item Utilities

**Editing Page Item Attributes in the Property Editor**

To edit item attributes in the Property Editor:


   Page Designer appears.

2. In either the Rendering tree or Grid Layout, select the item to edit.

   The Property Editor changes to display Property Editor - Page Item.

3. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

4. Edit the appropriate attributes.

   To learn more about a specific attribute on a page, click the attribute label. See "Viewing Field-Level Help" on page 1-21.

**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. View the item attributes in the Property Editor. See "Editing Page Item Attributes in the Property Editor" on page 15-12.

2. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

3. Locate and expand Default.

4. Under Default:

   a. Select a Default Type:

      - **Static Value**
Managing Page-Level Items in Page Designer

– PL/SQL Expression
– PL/SQL Function Body

b. In the field provided, enter a value or code 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.

**Configuring Page Item Security**

To configure page item security:

1. View the item attributes in the Property Editor. See “Editing Page Item Attributes in the Property Editor” on page 15-12.

2. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.


4. Configure the Security attributes described in Table 15–2.

**Table 15–2 Property Editor - Page Item, Security**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization Scheme</td>
<td>Select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Providing Security Through Authorization&quot; on page 20-49</td>
</tr>
</tbody>
</table>
Table 15–2 (Cont.) Property Editor - Page Item, Security

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session State Protection</td>
<td>Select a Session State Protection level. Options include:</td>
</tr>
<tr>
<td></td>
<td>- Unrestricted - The item may be set by passing the item in a URL or in a form. No checksum is required in the URL.</td>
</tr>
<tr>
<td></td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>data entry items and is always observed, even if Session State Protection is disabled. Use this attribute for application items or for page</td>
</tr>
<tr>
<td></td>
<td>items having the following Display As types: Display Only (Save State=No) and Text Field (Disabled, does not save state).</td>
</tr>
<tr>
<td></td>
<td>If Session State Protection is enabled, these additional attributes may be applied to items:</td>
</tr>
<tr>
<td></td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td>the workspace and application.</td>
</tr>
<tr>
<td></td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td>workspace, application, and user.</td>
</tr>
<tr>
<td></td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td>the current session.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Understanding Session State Protection&quot; on page 20-21</td>
</tr>
<tr>
<td>Store value encrypted in session state</td>
<td>Specify whether to encrypt this item when stored in session state. If the contents of an item contain sensitive data, then you should encrypt</td>
</tr>
<tr>
<td></td>
<td>the value when it is stored in the Application Express session state management tables. Otherwise, anyone with rights to read the Application</td>
</tr>
<tr>
<td></td>
<td>Express meta data tables could potentially write a query to extract this sensitive data. Values up to 4000 bytes in length can be encrypted.</td>
</tr>
<tr>
<td></td>
<td>Attempts to encrypt values longer than 4000 bytes produce an error message.</td>
</tr>
<tr>
<td>Escape Special Characters</td>
<td>This attribute only displays with some item types. Select Yes to prevent Cross-Site Scripting (XSS) attacks. Select No if you want that HTML</td>
</tr>
<tr>
<td></td>
<td>tags which are stored in the page item or in the entries of a list of value are actually rendered.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Understanding Cross-Site Scripting Protection&quot; on page 20-16</td>
</tr>
</tbody>
</table>
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 given item. Clicking on a quick pick sets the value of the item. The following example shows quick picks for 5%, 10%, 15%, and 20%.

To create a quick pick:

1. View the item attributes in the Property Editor. See "Editing Page Item Attributes in the Property Editor" on page 15-12.

2. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane.

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 X - Enter the label for each Quick Pick. This label is the text the end user of the application sees.
   d. Value X - 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.

Displaying Conditional Page Items

To display a conditional item:
1. View the item attributes in the Property Editor. See "Editing Page Item Attributes in the Property Editor" on page 15-12.

2. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.
   
   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane.

3. Expand Condition. Select a condition type and enter appropriate information in the field provided.

4. Click Save.

**Displaying Read Only Page Items**

To display a read-only item:

1. View the item attributes in the Property Editor. See "Editing Page Item Attributes in the Property Editor" on page 15-12.

2. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.
   
   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane.

3. Edit the appropriate attributes in the Property Editor.

4. Expand Read Only. Select a Read Only type and enter appropriate information in the field provided.

5. Click Save.

**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 appropriate application.

2. Click Utilities.

3. From Page Specific Utilities region, click Item 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

**Managing Page-Level Items in 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.
Managing Page-Level Items in Component View

- Viewing Page-Level Items in the Component View
- Creating Page-Level Items in Component View
- Editing Page-Level Items in Component View

See Also: "Understanding Page-Level Items" on page 15-1 and "Managing Page-Level Items in Page Designer" on page 15-7

Viewing Page-Level Items in the Component View

This section describes how to create and edit page-level items.

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" on page 15-27.
- **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" on page 15-23

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
- Creating a Static List of Values
- Creating a Cascading List of Values
- Creating Multiple Items Using a Tabular Form

See Also: "About the Differences Between Page Items and Application Items" on page 15-1 and "About Item Types" on page A-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 Component View" on page 9-5.

2. If necessary, create a region. See "About Regions" on page 14-4.

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" on 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" on 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" on page 15-2.
   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" on page 1-21.

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" on page 18-18

To create a static list of values:
1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
2. If necessary, create a region. See "About Regions" on page 14-4.
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" on 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" on page 15-2.
   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.
Managing Page-Level Items in Component View

- 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** 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
```

**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
```

**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
```

**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. Application Builder includes cascading LOV support for the following item types: Select List, Shuttle, Checkbox, Radio Group, Popup LOV, and List Manager. To learn more about item types, see "About Item Types" on 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 Component View" on page 9-5.
   b. If necessary, create a region. See "About Regions" on page 14-4.
   c. Under Items, click the Create icon.
      The Create Item Wizard appears.
   d. For Item Type, select Select List.
Managing Page-Level Items in Component View

15-21

For Display Position and Name:
- Item Name - Enter:
P1_DEPTNO
- Accept the remaining defaults and click Next.

For Item Attributes:
- Label - Enter Department
- Accept the remaining defaults and click Next.

For Settings, accept the defaults and click Next.

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.

Accept the remaining defaults.

Click Create Item.

2. Create the second item named P1_EMPNO:

a. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

b. If necessary, create a region. See "About Regions" on page 14-4.

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.

For Settings, accept the defaults and click Next.

For List of Values:
  - Display Null Value - Accept the default, Yes.
  - Null Display Value - Enter - Select Department -
Managing Page-Level Items in Component View

- 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.

See Also: "Editing Page-Level Items in Component View" on page 15-23

Creating Multiple Items Using a Tabular Form

To create multiple items simultaneously:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

2. If necessary, create a region. See "About Regions" on page 14-4.

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.

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
- Defining Default Values
- Configuring Page Item Security
- Creating a Quick Pick Selection
- Displaying Conditional Page Items
- Displaying Read Only Page Items
- Editing Multiple Items Simultaneously

See Also: "About Item Types" on page A-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 Component View" on page 9-5.
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" on page 1-21.
4. Click Apply Changes.

See Also: "Understanding Cross-Site Scripting Protection" on page 20-16

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:

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.

**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:


2. View the item attributes. Under Items, select the item name.

   The Edit Page Item page appears.


4. Configure the attributes described in Table 15–3.

**Table 15–3  Edit Page Item, Security**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization Scheme</td>
<td>Select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Providing Security Through Authorization&quot; on page 20-49</td>
</tr>
<tr>
<td>Session State Protection</td>
<td>Select a Session State Protection level. Options include:</td>
</tr>
<tr>
<td></td>
<td>■ Unrestricted - The item may be set by passing the item in a URL or in a form. No checksum is required in the URL.</td>
</tr>
<tr>
<td></td>
<td>■ 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. 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).</td>
</tr>
<tr>
<td></td>
<td>If Session State Protection is enabled, these additional attributes may be applied to items:</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Understanding Session State Protection&quot; on page 20-21</td>
</tr>
</tbody>
</table>
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%.
To create a quick pick:


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.

**Displaying Conditional Page Items**

To display a conditional item:


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.

**Displaying Read Only Page Items**

To display a read-only item:

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:
      ```html
      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**.

**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
- Reassigning Region Items
- Deleting Multiple Items
- Viewing Item Utilities
- Viewing Page Item History

**Editing Multiple Items** To edit multiple items simultaneously:

1. Navigate to the page containing the items. See "Viewing a Page in Component
   View" on page 9-5.

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" on
   page 15-23
The Page Items page displays items in an editable report. Table 15–4 describes editable attributes.

Table 15–4 Editable Attributes on Page Items

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>Specify the display sequence for this component. The sequence determines the order of evaluation.</td>
</tr>
<tr>
<td>Prompt</td>
<td>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.</td>
</tr>
<tr>
<td>Field Template</td>
<td>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.</td>
</tr>
<tr>
<td>Region</td>
<td>Defines the region in which the item displays. All items must be in a region.</td>
</tr>
<tr>
<td>New Line</td>
<td>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.</td>
</tr>
<tr>
<td>New Field</td>
<td>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.</td>
</tr>
<tr>
<td>Width</td>
<td>Specifies the length (in characters) of the form element that displays for this item.</td>
</tr>
<tr>
<td>Height</td>
<td>Specifies the height (in lines) for text areas and multi select lists.</td>
</tr>
<tr>
<td>Column Span</td>
<td>Items are laid out in HTML tables. This property defines the value to be used for the COLSPAN attribute in the table cell.</td>
</tr>
<tr>
<td>Row Span</td>
<td>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.</td>
</tr>
<tr>
<td>Post Element</td>
<td>Text or HTML code is rendered immediately after the page item element.</td>
</tr>
</tbody>
</table>

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 Component View" on page 9-5.

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.

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 Component View" on page 9-5.
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.

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 Component View" on page 9-5.
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

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 Component View" on page 9-5.
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.

Managing Dynamic Actions

This section provides an overview of dynamic actions and explains how to create and modify them.

- **About Dynamic Actions**
- **Viewing Plug-in Examples**
- **About Dynamic Action Events**
- **Managing Dynamic Actions in Page Designer**
- **Managing Dynamic Actions in Component View**
- **About Calling JavaScript Using a Dynamic Action**
- **Debugging Dynamic Actions**

**See Also:** "Accessing Page Specific Utilities" on page 7-65

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" on page 15-41 for information on how to debug problems.

The process of implementing a dynamic action involves the following:

1. Edit or create an 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.

For Dynamic Action implementation examples, go to the Learning Library at the following location, click the **All Content** tab and enter search criteria for Application Express (APEX) Product OBEs:

http://www.oracle.com/technetwork/tutorials/index.html

Viewing Plug-in Examples

For dynamic action implementation examples, go to the Learning Library at the following location, click the **All Content** tab and enter search criteria for Application Express (APEX) Product OBEs:

http://www.oracle.com/technetwork/tutorials/index.html
About Dynamic Action Events

You can define dynamic actions that can 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 (focusing)** - 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 **pageint**) – 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 mousedown 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.

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.
Managing Dynamic Actions

Managing Dynamic Actions in Page Designer

This section describes how to manage dynamic actions in Page Designer.

- Creating a Dynamic Action in Page Designer
- Editing Dynamic Actions in Page Designer
- Defining Dynamic Action Event Scope
- Deleting a Dynamic Action in Page Designer

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.

1. View the page to contain the item in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.

2. Click the Dynamic Actions tab on the left side of the page, beneath the Rendering tree.

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.

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. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

5. Under Property Editor - Dynamic Actions, 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. 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. See "About Dynamic Action Events" on page 15-31.

   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.

   **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.
e. Specify the name of the page element as follows:
   - 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.
   - 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.

Next, define the action that to be performed if the event evaluates to True or False.

6. 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.

7. 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.

8. Click Save.

**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 to contain the item in Page Designer. See "Viewing a Page in Page
Designer" on page 8-15.
2. Click the Dynamic Actions tab on the left side of the page. Dynamic action 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 - Dynamic Action, 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.

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" on page 15-37, "Adding Additional True Actions" on page 15-39, and "Defining the Frequency and Scope" on page 15-40

**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:


2. Click the **Dynamic Actions** tab on the left side of the page, beneath the Rendering tree.

3. Under Dynamic Actions, select the dynamic action.

   The Property Editor - Dynamic Action appears.

4. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

5. Under Advanced, select an Event Scope. Options include:

   - Static
   - Dynamic
   - Once

   **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**.
Deleting a Dynamic Action in Page Designer
To delete a dynamic action:

1. View the page to contain the item in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.
2. Click the Dynamic Actions tab on the left side of the page.
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.

Managing Dynamic Actions in Component View
This section describes how to manage dynamic actions in Component View.

- Creating a Dynamic Action in Component View
- Editing Dynamic Actions in Component View
- Adding Additional True Actions
- Defining the Frequency and Scope

See Also: "Managing Dynamic Actions in Page Designer" on page 15-34

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 Component View" on page 9-5.
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" on page 15-31.
   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" on page 1-21.

- **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” on page 1-21.

7. Click Create Dynamic Action.

### 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 Component View" on page 9-5.

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" on page 1-21.

4. Click Apply Changes.

   **See Also:** "Creating a Dynamic Action in Component View" on page 15-37, "Adding Additional True Actions" on page 15-39, and "Defining the Frequency and Scope" on page 15-40

### Adding Additional True Actions

This section describes how to add additional true actions.

- **About Adding Additional True Actions**
- **Adding a True Action**

**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.

**Adding a True Action** To add an action:
1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

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" on page 1-21.

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.

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 Component View" on page 9-5.

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.

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 and Set Value. You can also use JavaScript code for the condition of a dynamic action by setting the condition type to JavaScript Expression.

See Also: "Managing Dynamic Actions" on page 15-30

Debugging Dynamic Actions

This section describes how to debug dynamic actions in Oracle Application Express.

- About Debugging Dynamic Actions
- Debugging Dynamic Actions

About Debugging Dynamic Actions

Debugging dynamic actions in Oracle Application Express is slightly different than other debugging, because much of the processing done with the dynamic action framework is done on the client, not on the server. To debug dynamic actions, Oracle Application Express outputs debug information to the browser’s JavaScript console if your browser supports it (for example Firefox with Firebug installed shows the debug information in its Console pane). The debug information tells you when a dynamic action fires, the name of the dynamic action and also the Action JavaScript Object, which contains all the information needed by the client to execute the action (for example, event name, when details, action, affected elements and so on).

Debugging Dynamic Actions

To debug a dynamic action:
1. Ensure the application containing the dynamic action has Debugging enabled. See "Utilizing Debug Mode" on page 22-3.
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 see the following text output to the console:

```
Dynamic Action Fired: [Dynamic action name] {{(JavaScript Object containing all Dynamic Action information)}}
```

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.

## Creating Buttons

This section describes how to add button to your application. 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?
- Managing Buttons in Page Designer
- Managing Buttons in Component View
- About Calling JavaScript from a Button
- About the Relationship Between Button Names and REQUEST
- About Branching with Buttons

**See Also:** "Calling a Page from a Button URL" on page 2-20 and "Accessing Page Specific Utilities" on page 7-65

### 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).

### Managing Buttons in Page Designer

This section describes how to manage buttons in Page Designer.

- Creating a Button in Page Designer
- Editing a Button in Page Designer
- Displaying a Button Conditionally

**See Also:** "Managing Buttons in Component View" on page 15-44
Creating a Button in Page Designer
To create a button in Page Designer:

1. View the page to contain the item in Page Designer. See "Viewing a Page in Page Designer" on page 8-15.

2. If necessary, create a region to contain the item. See "About Regions" on page 14-4.

3. In the Gallery, click the Buttons tab.
   Passing the cursor over a button displays a tooltip that describes it.

4. Select a button and drag it to the appropriate location in Grid Layout.
   **Tip:** You also right-click the item, select Add To, and select the appropriate location.

   The Property Editor - Page Items appears.
   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. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

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.

Editing a Button in Page Designer
To create a region button in Page Designer:

   Page Designer appears.

2. In either the Rendering tree or Grid Layout, select the button to edit.
   The Property Editor changes to display Property Editor - Button.

3. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

4. Edit the appropriate attributes.
   To learn more about a specific attribute on a page, click the attribute label. See "Viewing Field-Level Help" on page 1-21.

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:

   Page Designer appears.
2. In either the Rendering tree or Grid Layout, select the button to edit. The Property Editor changes to display Property Editor - Button.

3. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane.

4. Expand Condition. Select a condition type and enter appropriate information in the field provided.

5. Click Save.

   **See Also:** "Referencing Session State Using Bind Variable Syntax" on page 2-24

### Managing Buttons in Component View

This section describes how to create buttons in Component View.

- Creating a Button in Component View
- Creating Multiple Buttons in Component View
- Editing a Button in Component View
- Displaying a Button Conditionally

**See Also:**

### 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 Component View" on page 9-5.

2. If necessary, create a region. See "About Regions" on page 14-4.

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" on page 1-21.
Creating Buttons

Managing Database Application Controls

15-45

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:

```html
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:

```html
	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" on page 1-21.

Action - Select a behavior when the button is clicked.

g. 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" on page 17-15.

h. 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" on page 15-49

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 Component View" on page 9-5.

2. If necessary, create a region. See "About Regions" on page 14-4.

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.

Editing a Button in Component View
This section describes how to edit a button in Component View.

- Editing Button Attributes
- About Execute Validations
- Editing Multiple Buttons at Once

Editing Button Attributes  To edit attributes for an existing button:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

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" on page 1-21.

4. Click **Apply Changes**.

**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" on
page 17-15

**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 Component View" on
   page 9-5.

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** To delete multiple buttons at once:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on
   page 9-5.

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**.

**History** Use the History page to view a summary of recent edits to buttons.
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" on page 15-44.
2. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
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.

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:

```javascript
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
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:

2. Create a dynamic action and using the action type Execute JavaScript Code to execute the previous code, for example:

```javascript
if (confirm("Would you like to perform this delete action?") { 
apex.submit('Delete');
```
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.

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 Application 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" on page 16-20

Creating Trees

This section describes how to create tree components in Application Builder.

- About Trees
- About Support for Legacy jsTrees
- Creating a Tree on New Page
- Managing Trees in Page Designer
- Managing Trees in Component View

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" on page 15-50.
About Trees

Application 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` clause to generate the hierarchical query.

**Tip:** The creation of APEX Tree regions is only supported on pages using a Desktop user interface.

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.

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 wizards generates this hierarchical query for your based on the options you select.

To create a tree on a new page:

1. On the Workspace home page, click the **Application 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.
   b. Parent ID - Select the column to use as the parent ID.
   c. Node Text - Select the text to appear on the tree nodes.
   d. Start With - Select the column to be used to specify the root of the hierarchical tree query.
   e. Start Tree - Choose how to start your query. Options include:
      - Based on Existing Item - Select an existing application or page item.
      - Based on a SQL Query - Enter a SQL query that returns a single row or single column.
      - Based on a Static Value - Enter a static value.
      - Value is NULL.
   f. Click Next.

9. For Where and Order by, specify the following:
   a. Where Clause - Enter an optional WHERE clause. For example:
      `ename='JONE'`
   b. Order Siblings By - Select the order siblings by column, such as ENAME. The default value is based on the Node Text column selected.
   c. Click Next.

10. For Tree Attributes, specify the following:
    a. Include Buttons - Select the buttons to include.
    b. Selected Node Page Item - Select the page or application item to hold the selected node value. This item can be used to save the tree state, by holding the value of the last selected node. The value of the selected node can be saved
to the selected item using the node link attribute or a page process. When the tree is reloaded, the tree opens to the last selected tree node.

c. Tooltip - Displays a tooltip when the mouse hovers over a leaf node. Options include:
   - Static Assignment - Specifies a static value to be used as the tooltip text. Substitution strings can be used in the static text, to incorporate information from the SQL query in the tooltip text. The following substitution strings can be used:
     #VALUE# - refers to the value of ID column.
     #TITLE# - refers to the value of Node Text column
     #LINK# - refers to the value of Link option
   - Database Column - Select the column to use as the tooltip text.

d. Link Option:
   - Existing Application Item - Makes the leaf node text a link. If you select this option, you must specify a page to link to and an existing application item to link leaf node text.
     This option only appears if you select the Existing Application Item link option.
   - Nothing

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.

Managing Trees in Page Designer

This section describes how to manage trees in Page Designer.

- Creating a Tree in Page Designer
- Editing Tree Attributes in Page Designer

Creating a Tree in Page Designer

To create a tree on an existing page:


Page Designer appears

2. From the Gallery select the Tree region and drag it to the appropriate location in Grid Layout.

   Tip: You also right-click Tree 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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

4. Edit the region 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** - Enter a title for the region.

   b. **Source** - Enter the SQL query underlying the tree. The query must be in the following format:

   ```sql
   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:

   ```sql
   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#go-small.png'.

   c. **Layout** - Specify the Sequence, Parent Region, and Position.
d. Appearance - Specify the template.

5. Select the region Attributes in the Rendering tree.
The Property Editor displays attributes for the tree.

6. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

7. Edit the tree attributes:
   a. Tree Implementation - Select how the tree should be implemented, either APEX Tree or jsTree (legacy).
   b. Tree Template - Select the template to apply to the tree from the available jsTree template options. This option is only applicable to jsTree regions.
   c. Activate Node Link With - Select the action required to activate node links on your tree.
   d. Selected Node Page Item - Select the page or application item that you wish to hold the selected node value.
   e. Tooltip - Select whether tooltips are displayed, and the source for the tooltip.
   f. Static Tree ID - Enter the unique identifier for the tree.
   g. 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.

8. Click Save.

Editing Tree Attributes in Page Designer
By default, a tree does not have focus and node links are activated with a single click action. This section describes how developers can customize a tree by editing tree attributes.

To access the Tree attributes page:
   Page Designer appears.
2. Select the region in the Rendering tree or Grid Layout.
   The Property Editor displays attributes for the tree region.
3. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.
4. Edit the region attributes:
   Tip: To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane.

5. Click Save.

Managing Trees in Component View
This section describes how to manage trees in Component View.
- Creating a Tree in Component View
- Editing Tree Attributes in Component View
Creating a Tree in Component View

To create a tree in component view:

1. Navigate to the page. See "Viewing a Page in Component View" on page 9-5.

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.
   f. Click Next.

7. For Where and Order by, specify the following:
   a. Where Clause - Enter an optional WHERE clause.
   b. Order Siblings By - Select the columns by which siblings are ordered.
c. Click Next.

8. For Tree Attributes, specify the following:
   a. Include Buttons - Select the buttons to include.
   b. Selected Node Page Item - Can be used to hold the selected node value. If you select an application item, you can use this item to save the tree state, by holding the last selected node of the tree.
   c. Tooltip - Displays a tooltip when the mouse hovers over a leaf node. Options include:
      - Static Assignment - Specifies a static value to be used as the tooltip text. Substitution strings can be used in the static text, to incorporate information from the SQL query in the tooltip text. The following substitution strings can be used:
         
         #VALUE# - refers to the value of ID column.
         #TITLE# - refers to the value of Node Text column
         #LINK# - refers to the value of Link option
      - Database Column - Select the column to use as the tooltip text.
   d. Link Option:
      - Existing Application Item - Makes the leaf node text a link. If you select this option, you must specify a page to link to and an existing application item to link leaf node text.
      This option only appears if you select the Existing Application Item link option.
      - Nothing
   e. Click Next.

9. Confirm your selections and click Create Region.

Editing Tree Attributes in Component View

By default, a tree does not have focus and node links are activated with a single click action. This section describes how developers can customize a tree by editing tree attributes.

To access the Tree Attributes page:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

2. Access the Region Attributes page. Under Regions, click the Tree link next to the name of the tree region you want to edit.

   Edit the appropriate attributes.

   **Tip:** To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

3. Click Apply Changes.
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" on page 18-35 for information about referencing a shortcut inside of a JavaScript literal string, "Understanding Validations" on page 17-6, and "About Using JavaScript Code in a Plug-In" on page 19-14
When you build a database application, you can include different types of navigation controls. The available navigation option 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.

- About Creating Lists as Navigation
- Creating Tabs
- Creating Breadcrumbs
- Creating a Navigation Bar Entry
- Controlling Navigation Using Branches

**See Also:** "Creating Trees" on page 15-49

### 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 list that functions as navigation using the template Vertical Unordered List with Bullets. Newer themes, such as *Universal Theme* - 42, supports 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" on page 18-20, "About Navigation Menus" on page 18-21, and "Understanding Sample Database Application" on page 3-6

### 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.
Creating Tabs

Tip: As an alternative to tabs, you can use lists to display tab controls. List templates provide greater control over HTML generation. See "Creating Lists" on page 18-20 and "About Navigation Menus" on page 18-21.

- About Tabs
- About the Tabs Page
- Checking for Tab Template Support
- Creating a Tab
- Editing Tabs
- Accessing Tab Reports

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.

See Also: "Creating a Database Application" on page 7-4

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. Application 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 "Understanding Sample Database Application" on page 3-6.

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 Application 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.

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" on page 16-3 and "Understanding Sample Database Application" on page 3-6

### 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 **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

3. Locate **Page Templates** at the bottom of the page. If needed, expand the Page Templates region.

<table>
<thead>
<tr>
<th>Page Template Name</th>
<th>Supports Standard Tabs</th>
<th>Supports Parent Tabs</th>
<th>Template Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>No</td>
<td>No</td>
<td>Login</td>
</tr>
<tr>
<td>No Tabs - Left Sidebar (fixed-width / DIV based)</td>
<td>No</td>
<td>No</td>
<td>No Tabs with Sidebar</td>
</tr>
<tr>
<td>No Tabs - Left and Right Sidebar (fixed-width / DIV left and optional table-based right)</td>
<td>No</td>
<td>No</td>
<td>No Tabs with Sidebar</td>
</tr>
<tr>
<td>No Tabs - Right Sidebar (fixed-width / DIV based)</td>
<td>No</td>
<td>No</td>
<td>No Tabs</td>
</tr>
<tr>
<td>No Tabs - Right Sidebar (optional / table-based)</td>
<td>No</td>
<td>No</td>
<td>No Tabs</td>
</tr>
<tr>
<td>One Level Tabs - Left Sidebar (fixed-width / DIV based)</td>
<td>Yes</td>
<td>No</td>
<td>One Level Tabs with Sidebar</td>
</tr>
<tr>
<td>One Level Tabs - Left and Right Sidebar (fixed-width / DIV left and optional table-based right)</td>
<td>Yes</td>
<td>No</td>
<td>One Level Tabs with Sidebar</td>
</tr>
</tbody>
</table>

**See Also:** "Page Templates" on page 13-42 and "Using Themes" on page 13-1 for information about setting a default page template at the application level

### Creating a Tab

An application can have pages with no tabs, one level of tabs, and two levels of tabs. Standard tabs enable you to display only one level of tabs. To display two levels of tabs, you must define both Parent tabs and Standard tabs.

To create a tab:

1. Access the Tabs page:
   a. On the Workspace home page, click the **Application 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" on page 16-2

**Editing Tabs**

This section describes how to edit tabs. You can edit multiple tabs simultaneously. Also, for standard tabs, you can update tab properties, such as their labels and order sequence, by using the Standard Tab Tasks list.

- Editing Multiple Tabs Simultaneously
- Managing Standard Tabs
Creating Tabs

Editing Multiple Tabs Simultaneously
To edit multiple tabs simultaneously:

1. Navigate to the Tabs page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
   d. Under Navigation, click Tabs.
2. Click one of the following tabs at the top of the page:
   ■ Edit Standard Tabs
   ■ Edit Parent Tabs
   A report appears.
3. To edit a specific tab, click the Edit icon.
4. Edit the appropriate attributes and click Apply Changes.

Managing Standard Tabs
The Standard Tab Task list displays on the right side of the Tabs page. You can access the links on this list to move a standard tab to different parent tab, rename a standard tab set, resequence the display order, associate pages with a tab set, create a new standard tab, or create a new standard tab set.

To access the Standard Tab Task list:
1. Navigate to the Tabs page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
   d. Under Navigation, click Tabs.
2. Click Manage Tabs.
3. Make a selection from the Standard Tab Task list on the right side of the page:
   ■ Move Highlighted Standard Tab to a Different Parent Tab
   ■ Rename Standard Tab Set
   ■ Resequence Display Order
   ■ Associate Page(s) with Selected Standard Tab
   ■ Create New Standard Tab
   ■ Create New Standard Tab Set

Accessing Tab Reports
This section describes the Conditional Display, Utilization, and History reports.

To view the Conditional Display report:
1. Navigate to the Tabs page:
   a. Navigate to the Workspace home page.
Creating Breadcrumbs

This section explains how to create and manage 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?
- Creating a Breadcrumb While Creating a Page
- Editing Breadcrumbs
- Reparenting Breadcrumb Entries
- Deleting Unused Breadcrumb Entries
- Accessing Breadcrumb Reports

See Also: "Creating a Theme" on page 13-15 and "Breadcrumb Templates" on page 13-33

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.
Creating Breadcrumbs

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. See "Managing Pages in a Database Application" on page 7-36.

   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.

Creating a Breadcrumb Manually

This section describes how to create breadcrumbs manually.

- About Creating a Breadcrumb Manually
- Creating Breadcrumbs
- Adding Breadcrumb Entries
- Adding Breadcrumb Regions
- About Creating Dynamic Breadcrumbs

See Also: "Understanding Sample Database Application" on page 3-6
About Creating a Breadcrumb Manually
To create breadcrumbs manually, you must add a breadcrumb to each page in your application as follows:

- **Step 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. See "Creating Breadcrumbs" on page 16-9.

- **Step 2:** Add entries to the breadcrumb. See "Adding Breadcrumb Entries" on page 16-10.

- **Step 3:** Add the breadcrumb to a page by creating a region. See "Adding Breadcrumb Regions" on page 16-11.

Creating Breadcrumbs
This section describes the various way to create breadcrumbs.

- Creating Breadcrumbs as Shared Components
- Creating a Breadcrumb from Page Designer
- Creating Breadcrumbs from Component View

Creating Breadcrumbs as Shared Components To create breadcrumbs from the Shared Components page:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Breadcrumbs page appears.
5. Click Create Breadcrumb.
6. Enter a name and click Create.

Creating a Breadcrumb from Page Designer To create a breadcrumb from Page Designer:

2. On the Page Designer Toolbar, click the Create menu and select Shared Components.
3. On Create Application Component, select Breadcrumb and click Next.
4. InBreadcrumb name, enter a name for this breadcrumb and click Create.
5. Once you create a named breadcrumb, you add breadcrumb entries. See "Adding Breadcrumb Entries" on page 16-10.

Creating Breadcrumbs from Component View To create breadcrumbs from a Component View:

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 "Adding Breadcrumb Entries" on page 16-10.

Adding Breadcrumb Entries

To add a breadcrumb entry:

1. Navigate to the Breadcrumbs page:
   a. Navigate to the Workspace home page.
   b. Click the **Application Builder** icon.
   c. Select an application.
   d. Click **Shared Components**.
   e. Under Navigation, click **Breadcrumbs**.

   The Breadcrumbs page appears.

2. Select a breadcrumb to which to add entries.

3. Click **Create Breadcrumb Entry**.

4. Under Breadcrumb, select the page where this breadcrumb entry displays.

5. Under Entry:
   a. Sequence - Indicate the order in which breadcrumb entries appear. The sequence determines the order of evaluation.
   b. Parent Entry - Identify the parent of this entry. Any given breadcrumb should have only one root node, identified as an item with no parent ID.
   c. Short Name - Identify the short name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.
   d. Long Name - Identify the long name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.

6. Under Target, specify the target location:
   a. Target is a - Select **Page in this Application**.
   b. Page - Specify the target page number.

   To reset pagination for that page, select **reset pagination for this page**.
   c. Request - Enter text for which you would like to set the built-in application item called REQUEST. This is also the item that is set with the name of a button that was clicked.
   d. Clear Cache - Enter the page numbers (separated by commas) for those pages for which you would like to clear the user’s session state.
   e. To set session state (that is, give a listed item a value):
      - Set these items - Enter the list of application item names (separated by commas) for which you would like to set the session state (give the listed item a value).
      - With these values - Enter a comma delimited list of values for the items specified in Set these items. You can specify static values or substitution syntax (for example, &APP_ITEM_NAME.).
You can specify static values or substitution syntax (for example, &APP_ITEM_NAME). Note that item values passed to f?p= in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, \1234, 56\).

7. For Conditions:
   a. Condition Type - Select a condition type from the list. This condition must be met in order for this component to be rendered or processed.
   b. Enter an expression in the fields provided.

8. For Authorization, optionally select an authorization scheme. This authorization scheme must evaluate to TRUE in order for this component to be rendered or otherwise processed.

9. For Configuration, select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.

10. Click Create Breadcrumb Entry.
Repeat these procedures for each breadcrumb entry you create.

Adding Breadcrumb Regions
A region is an area on a page that serves as a container for content. Once you create a breadcrumb and a breadcrumb template, the next step is to create a region. Once you create a breadcrumb region, you can add a breadcrumb to a page.

- Adding a Breadcrumb Region in Page Designer
- Adding a Breadcrumb Region in Component View

See Also: "Creating a New Template" on page 13-26 and "Breadcrumb Templates" on page 13-33 for information about changing menu display

Adding a Breadcrumb Region in Page Designer
To create a breadcrumb region in Page Designer:

   Page Designer appears. 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. From the Gallery, Regions, select Breacrumb drag it to the appropriate location in Grid Layout.
   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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.
   Required attributes display a red triangle in the upper left corner next above attribute label.

4. Edit the appropriate attributes in the Property Editor.
   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.
   
   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 grid 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.
   e. Appearance, Item Display Position - Specify where page items display in relation to the main region content.

5. Click **Save**.
   Repeat these procedures for each page where you would like to add breadcrumb navigation.

**Adding a Breadcrumb Region in Component View**  To create a breadcrumb region in Component View:

1. Navigate to the appropriate page. See “**Viewing a Page in Component View**” on page 9-5.
2. Under Regions, click the **Create** icon.
   The Create Region Wizard appears.
3. For the region type, select **Breadcrumb** and click **Next**.
4. ForBreadcrumb 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. ForBreadcrumb:
   a. Breadcrumb - Select the breadcrumb to be associated with this region.
   b. Breadcrumb Template - Select a template.
c. Click Next.

6. Review the breadcrumb region information and click Next.

7. Click Finish.

Repeat these procedures for each page where you would like to add breadcrumb navigation.

About Creating Dynamic Breadcrumbs

To give users more exact context, you can include session state in breadcrumbs, making your breadcrumbs dynamic. For example, suppose a page in your application displays a list of orders for a particular company and you want to include the following breadcrumb:

Home > Orders > Orders for ACME Inc

In this example, ACME Inc not only indicates the page a user is on but also the navigation path. The Application Express engine stores the value of ACME Inc. in session state.

To create this type of dynamic menu, you must include a reference to a session state item in the breadcrumb's short name or long name. For example:

&COMPANY_NAME.

Editing Breadcrumbs

Once you create a breadcrumb, you can edit it on the Breadcrumbs page.

To edit a breadcrumb:

1. Navigate to the Breadcrumbs page:
   a. Navigate to the Workspace home page.
   b. Click the Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
   e. Under Navigation, select Breadcrumbs.

   The Breadcrumbs page appears.

2. Select a breadcrumb.

3. Select a breadcrumb entry.

   The Create/Edit page appears.

4. Edit the appropriate attributes.

   To learn more about a specific item, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

5. (Optional) In the Breadcrumb Entry list on the right side of the page, you can select the Synchronize Breadcrumb With Page Name and Title option.

   Selecting this option makes changing the name of a page and breadcrumb a one-step process. The information you provide for breadcrumb names is used to update the referenced page name and title.

6. Click Apply Changes.
Creating Breadcrumbs

See Also:  "Accessing Breadcrumb Reports" on page 16-15

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 Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
   e. Under Navigation, select Breadcrumbs.
      The Breadcrumbs page appears.

2. Select a breadcrumb.
   The Entries page appears.

3. From the Tasks list, click Reparent Entries within this Breadcrumb.
   The Reparent Entries page appears.

4. Use the navigation bar to edit or filter the view:
   ■ Breadcrumb - Identify the breadcrumb you want to edit and click Go.
   ■ Start With - Make a selection to restrict your view to a subset of the breadcrumb hierarchy and click Go.

5. From Reparent to, select the new parent.

6. Select the breadcrumbs entries you want to move and click Reparent Checked Entries.

Deleting Unused Breadcrumb Entries
To delete unused breadcrumb entries:

1. Navigate to the Breadcrumbs page:
   a. Navigate to the Workspace home page.
   b. Click the Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
   e. Under Navigation, select Breadcrumbs.
      The Breadcrumbs page appears.

2. Select a breadcrumb.
   The Breadcrumb Entries page appears.

3. From the Tasks list, click Delete Unused Breadcrumb Entries.
   The Delete page appears.

4. Select the entries to remove and click Delete Checked.
Accessing Breadcrumb Reports
You can view the Breadcrumb Hierarchy, Grid Edit Breadcrumb, Breadcrumb Exceptions, Breadcrumb Utilization and Breadcrumb History reports by clicking the appropriate tab at the top of the Breadcrumbs page.

Note: These reports only appear after you create a breadcrumb.

To view Breadcrumb reports:
1. Navigate to the Breadcrumbs page:
   a. Navigate to the Workspace home page.
   b. Click the Application 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 "Customizing Interactive Reports as an End User" on page 10-57.

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.

See Also: "Creating Custom Themes" on page 13-23
Creating a Navigation Bar Entry

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" on page 3-6

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
- Creating a Navigation Bar Entry for Feedback

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 Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
2. Click Create.
   The Create Navigation Bar Entry Wizard appears.
3. For Method, select From Scratch.
4. For Type, select **Navigation to URL**.

   **Tip:** To learn more about Feedback, see "Creating a Navigation Bar Entry for Feedback" on page 16-18.

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. See "Viewing Field-Level Help" on page 1-21.
   
   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. "Editing Themes" on page 13-8 and "Optimizing a Page for Printing" on page 14-2.
      
      – 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 ?p= in the URL cannot con-
Creating a Navigation Bar Entry

tain 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" on page 13-42.

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 Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.

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.

See Also: "Managing Feedback" on page 5-16

Tip: In order for a navigation bar to appear, your page template must include the #NAVIGATION_BAR# substitution string. See "Page Templates" on page 13-42.

Copying a Navigation Bar Entry

To copy a navigation bar entry:

1. Navigate to the Navigation Bar Entries page:
Creating a Navigation Bar Entry

a. Navigate to the Workspace home page.
b. Click the Application Builder icon.
c. Select an application.
d. On the Application home page, click Shared Components.

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.

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.
   b. Click the Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.

   The Navigation Bar Entries page appears.

2. Select a navigation bar entry.

   The Edit page appears.

3. Edit the appropriate attributes.

4. To learn more about a specific item on a page, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

5. Click Apply Changes.

Editing Multiple Navigation Bar Entries Simultaneously

To edit multiple navigation bar entries simultaneously:

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
2. Click Grid Edit at the top of the page.
3. Edit the appropriate attributes and click Apply Changes.

Accessing Navigation Bar Entry Reports
You can view the Navigation Bar Entry Subscription and Navigation Bar Entry History reports by clicking the appropriate tab at the top of the Navigation Bar Entries page.

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
2. Click the appropriate tab:
   ■ Subscription - Click Subscription to access the Subscribed NavBars report. This report displays subscribed navigation bar entries in your application.
   ■ History - Click History to view the Navigation Bar History report. This report lists recent changes to navigation bars.

Controlling Navigation Using Branches
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
To create a branch in Page Designer:
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click Branches and select Create Branch.

   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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

4. For Identification, Name - Specify the name of the branch for easy identification by developers.

5. For Execution Options:
   a. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
   b. Point - Select the point in the processing at which this branch action is considered.

6. For Behavior, Type - Select a page branch type.
   The attributes that display in the Property Editor depend on the branch type you select. To learn more, click the Help tab.
   If Type is **Page or URL (Redirect)** - Click the **Target** field and edit the attributes in the Link Builder Target dialog.

7. For 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.

8. Click **Save**.

   **See Also:** "About the When Button Pressed Attribute" on page 2-39, "Creating Pre-Rendering Branches, Computations, and Processes" on page 8-17, and "Creating Page Submission Branches, Validations, Computations, and Processes" on page 8-18

**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 Component View" on page 9-5.

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” on page 2-39

### About Branching Conditionally

Like other controls, branches can be made conditional. To create a conditional branch, make a selection from the Condition Type list, and enter text in the expression fields to implement the condition type you choose.
This section describes how to manage computations, validations, and processes.

- Understanding Page Computations
- Understanding Validations
- Understanding Page Processes
- About Branches
- Understanding Application Processes
- Understanding Application Computations
- Using the Attribute Dictionary

See Also: "Quick Start" on page 1-1, "Application Builder Concepts" on page 2-1, "Creating Database Applications" on page 7-1, "Controlling Page Layout" on page 14-1, and "Adding Navigation" on page 16-1

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
- Managing Page Computations in Page Designer
- Managing Page Computations in Component View

See Also: "Understanding Application Computations" on page 17-30

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.
Managing Page Computations in Page Designer

This section describes how to create page computations in Page Designer.

- **Tip:** To learn more about execution behavior in the case of a validation error, see "What Happens When a Validation Fails?" on page 17-17.

- Creating a Page Computation in Page Designer
- Computation Points and Computation Syntax Example
- Editing Page Computation Attributes in Page Designer
- About Editing the Computation Point and Source
- About Creating Conditional Computations

See Also: "Understanding Application Computations" on page 17-30

Creating a Page Computation in Page Designer

To create a page computation:


2. In the Rendering tree, expand the Pre-Rendering or Post-Rendering node and locate where the computation should be.

3. Right-click Computations and select **Create Computation**.

Attributes for the computation 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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

5. Under Property Editor - Computations, 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. 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.

   Next, define the action that to be performed if the event evaluates to True or False.

6. Click **Save**.
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. See "Viewing a Page in Page Designer" on page 8-15.
2. In the Rendering tree, 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.

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" on page 17-4

About Creating Conditional Computations
You can make a computation conditional by making a selection from the Condition, Type list and populating the appropriate fields.

**See Also:** "Understanding Conditional Rendering and Processing" on page 2-12

Managing Page Computations in Component View
This section describes how to create 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?" on page 17-17

- Creating a Page Computation in Component View
- Creating a Page Computation in Component View
- Editing Page Computation Attributes in Component View
About Creating Conditional Computations

See Also: "Understanding Application Computations" on page 17-30 and "Utilizing Cross Page Utilities" on page 7-66

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 Component View" on page 9-5.

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:
   - Item on This Page
   - Item on Another Page
   - 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.

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 Component View" on page 9-5.

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.

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" on page 17-4

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" on page 17-4 and "Understanding Conditional Rendering and Processing" on page 2-12

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 Component View" on page 9-5.

2. Create an item named P10_PHONE_NUMBER to store the combined values of P10_AREA_CODE, P10_PREFIX, and P10_LINE_NUMBER. See "About the Differences Between Page Items and Application Items" on page 15-1.

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.
Understanding Validations

This section describes how to create validations.

- About Validations
- Managing Validations in Page Designer
- Managing Validations in Component View
- About Determining When Validations Execute
- About Defining How Validation Error Messages Display
- Processing Validations Conditionally
- About Altering the Go to Error Link
- What Happens When a Validation Fails?
About Validations

A validation is an edit check. Validations specific to a single item are page item validations. Validations that apply to an entire page are page validations. Validations for tabular forms specific to a single column are column level validations. Validations for tabular forms that do not apply to a single column are tabular form row validations.

You can define a validation declaratively by selecting a validation method. You enter the actual validation edit check in the Validation Messages field. Be aware that if a validation fails, subsequent page processes or computations do not occur. Also remember that the validation you enter must be consistent with the validation type you selected.

Managing Validations in Page Designer

This section describes how to create validations.

- Creating a Validation in Page Designer
- Editing a Validation in Page Designer

See Also: "Validating User Input in Forms" on page 11-21, "About Execute Validations" on page 15-47, and "About Item Types" on page A-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.

---

2. Click the Processing tab on the left side of the page, beneath the Rendering tree.
3. Expand Validating. Right-click Validations and select Create Validation. Validation 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.

4. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.
5. Under Property Editor - Validations, 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. 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, 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. To learn more, click the Help tab.

e. Validation, Always Execute - Specify whether this validation always executes.

f. Error, Error Message - Enter text for the error message.

g. Error, Display Location - Select where the error message displays for this validation.

h. Error, Associated Item/Column - Select the item or column where this validation error message displays.

6. Click Save.

See Also: "Creating Pre-Rendering Branches, Computations, and Processes" on page 8-17 and "Creating Page Submission Branches, Validations, Computations, and Processes" on page 8-18

Editing a Validation in Page Designer

To edit a validation in Page Designer:


2. Click the Processing tab on the left side of the page, beneath the Rendering tree.

3. Expand Validating and then Validations and select the validation. Validation 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.

4. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

5. Under Property Editor - Validations, edit the appropriate attributes:

   Tip: To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the center pane.

6. Click Save.

Managing Validations in Component View

You 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
- Creating a Validation for an Entire Page
- Creating a Validation for a Tabular Form Column
- Creating a Validation for a Tabular Form Row
- Editing a Validation in Component View

**See Also:** "Validating User Input in Forms" on page 11-21, "About Execute Validations" on page 15-47, and "About Item Types" on page A-1

**Creating a Validation for a Page Item**

To create a validation for a page item:

**Table 17-1  Page Item Validation Methods**

<table>
<thead>
<tr>
<th>Validation Method</th>
<th>Descriptions</th>
</tr>
</thead>
</table>
| Not Null          | Checks if an item's value in session state is null.  
|                   | 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. |
### String Comparison

String Comparison compares the value of an item to a specific string. When prompted select a validation type.

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.

In Expression 1, enter the name of the item you want to validate without a colon. For example:

```
P1_VALUE
```

In Expression 2, enter a string of values you want to validate against. For example:

```
ABC/DEF/GHI
```

### Regular Expression

Regular expressions provide a method to describe text patterns. Use a Regular Expression validation to perform data validation. When prompted, enter a regular expression.

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:

```
^\((?:[[:digit:]]{6},)+\)$
```

This regular expression would find the following entries valid:

- `123456,654321`
- `123456`
- `123456,123456,654321`

However, the following would not be valid:

- `123456,12345`
- `12345`

### SQL

SQL compares to values in the database. Select a validation type:

- **Exists**
- **NOT Exists**
- **SQL Expression**

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`.

```
SELECT 1 FROM customers
WHERE last_name = :P1_LAST_NAME
```
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.

      To learn more, see "About Determining When Validations Execute" on page 17-15.
   
   c. Click Next.

8. Define conditions that apply to this validation and click Create.

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 Component View" on page 9-5.
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 17–2 and click Next.

Table 17–2 Page Validation Methods

<table>
<thead>
<tr>
<th>Validation Method</th>
<th>Descriptions</th>
</tr>
</thead>
</table>
| SQL               | Compares to values in the database. Select a validation type:  
|                   |   ■ Exists  
|                   |   ■ NOT Exists  
|                   |   ■ SQL Expression  
| PL/SQL            | Useful if you need complex logic. Select a validation type:  
|                   |   ■ PL/SQL Expression  
|                   |   ■ PL/SQL Error  
|                   |   ■ Function Returning Boolean  
|                   |   ■ Function Returning Error Text  

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.
      To learn more, see "About Determining When Validations Execute" on page 17-15.
   d. Click Next.

7. Define conditions that apply to this validation and click Create.
Creating a Validation for a Tabular Form Column

To create a validation for a tabular form column:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.

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.

      To learn more, see "About Determining When Validations Execute" on page 17-15.
   d. Click Next.

8. For Conditions, define conditions that apply to this validation.
9. Click Create Validation.

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 Component View" on page 9-5.

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.
   - 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.

5. For Validation Type, select a validation type and follow the on-screen instructions and click Next.

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 #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.
     To learn more, see "About Determining When Validations Execute" on page 17-15.
   - d. Click Next.

7. For Conditions:
a. When Button Pressed - Select a page buttons which when selected permits this validation to be performed, subject to other conditions.

b. Condition Type - Select a condition type from the list that must be met in order for this component to be rendered processed.

c. Click Next.

8. Click Create Validation.

**Editing a Validation in Component View**

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 Component View" on page 9-5.

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. See "Viewing Field-Level Help" on page 1-21.

4. Click Apply Changes.

**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" on page 17-8 and "Editing a Validation in Component View" on page 17-15
About Defining How Validation Error Messages Display

You can choose to have validation error messages display inline (that is, on the page where the validation is performed) or on a separate error page.

You define how a validation error message displays by configuring the following validation attributes:

1. Error Message - Enter the error message text in this field.
2. Display Location - Select where the error message displays for this validation. Validation error messages display on a separate error page, or inline with the existing page. Inline error messages display underneath the Associated Item label and/or in a Notification area, defined as part of the page template.

   **Tip:** From Component View, this attribute is named **Error Message display location**.

Options include:

- **Inline with Field and in Notification** - Error displays inline within the field label and within a notification area (defined as part of the page template).
- **Inline with Field** - Error displays inline within the field label.
- **Inline in Notification** - Error displays inline within a notification area (defined as part of the page template).
- **On Error Page** - Displays the error on a separate error page. To create a hard error that stops all processing (including validations), you must display the error on an error page.

3. Associated Item (or Column) - Select the item (or column) where this validation error message displays.

   If you select an item and the Error message display location includes **Inline with Field**, the error message displays underneath the label of the specified item. If the Error message display location does not include **Inline with Field**, this attribute has no impact on the display of the error message.

**See Also:** "Editing a Validation in Page Designer" on page 17-8 and "Editing a Validation in Component View" on page 17-15

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" on page 17-8, "Editing a Validation in Component View" on page 17-15, and "Understanding Conditional Rendering and Processing" on page 2-12

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. To learn more, see "Translating Messages Used Internally by Oracle Application Express" on page 21-14.

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 they could potentially alter the data entered by the user. For example, an Automated Row Fetch process would fetch and overwrite the modified data with the values from the database when the user actually wants to get the entered data in order to fix the validation error.

If a computation is skipped, the following displays in the debug output:

Do not perform computation because inline validation errors found.

If an application or page process is skipped, the following displays in the debug output:

Skip because inline validation errors found.

Understanding Page Processes

This section describes how to create a page process.

- About Page Processes
- Managing Page Processes in Page Designer
- Managing Page Processes in Component View
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" on page 17-25 and "Utilizing Cross Page Utilities" on page 7-66

Managing Page Processes in Page Designer

This section describes how to create a page process.

- Creating Page Processes in Page Designer
- Editing Page Process Attributes in Page Designer
- Controlling When a Page Process Executes in Page Designer
- Creating Page Process Success and Error Messages in Page Designer
- Making a Page Process Conditional in Page Designer

**See Also:** "Understanding Application Processes" on page 17-25 and "Utilizing Cross Page Utilities" on page 7-66

**Tip:** To learn more about execution behavior in the case of a validation error, see "What Happens When a Validation Fails?" on page 17-17.

Creating Page Processes in Page Designer

To create a page process:


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. View all attributes in the Property Editor. Select **Show All** from the Property Editor Menu.

   **Tip:** To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

4. Under Identification:
   a. Name - Enter the name of the process.
b. Type - Specify which process type to execute. The Type you select determines what Setting attributes display.

5. Under Settings, edit the appropriate attributes. Required attributes display a red triangle in the upper left corner next above attribute label.

6. Under Execution Options:
   a. Sequence - Specify the sequence for this computation. The sequence determines the order of execution.
   b. Point - Select the point at which this process is executed.
   c. Run Process - Select the frequency that this process is executed.

7. Under Success Message:
   a. Error Message - Enter the success message for this process. If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed.

8. Under Error:
   a. Error Message - Enter the error message for this process. This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays.
   b. Display Location - Select where this process error message displays, when the process fails to complete successfully.

9. Click Save.

See Also: "Creating Pre-Rendering Branches, Computations, and Processes" on page 8-17 and "Creating Page Submission Branches, Validations, Computations, and Processes" on page 8-18

Editing Page Process Attributes in Page Designer
To edit an existing page process:

2. In either the Rendering or Processing tab, expand the nodes to locate the process and then select it.

   Process attributes display in the Property Editor.

3. View all attributes in the Property Editor. Select Show All from the Property Editor Menu.

4. Under Property Editor - Process, 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.

5. Click Save.

Controlling When a Page Process Executes in Page Designer
To control when a process executes, edit the following attributes:

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.

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:


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: "Page Templates" on page 13-42

Making a Page Process Conditional in Page Designer

To create a conditional page process, edit the following attributes:


2. Edit the following attributes:
   a. Condition, 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. Condition, 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. Condition, 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.

Managing Page Processes in Component View
This section describes how to create a page process.

- Managing Page Processes in Component View
- Creating Page Processes in Component View
- Editing Page Process Attributes in Component View
- Controlling When a Page Process Executes
- Creating Page Process Success and Error Messages
- Making a Page Process Conditional

See Also: "Understanding Application Processes" on page 17-25 and "Utilizing Cross Page Utilities" on page 7-66

Tip: To learn more about execution behavior in the case of a validation error, see "What Happens When a Validation Fails?" on page 17-17

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
- Creating a Tabular Form Page Process

Creating a Page Process To create a page process:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.


3. For Process Type, select the category of the process you want to create as described in Table 17–3.
Understanding Page Processes

Creating a Tabular Form Page Process

4. Follow the on-screen instructions.

4. Follow the on-screen instructions.

Creating a Tabular Form Page Process

To create a tabular form page process:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.


3. For Process Type:
   - Tabular Form - Select the associated tabular form region.

---

**Table 17–3 Page Process Types**

<table>
<thead>
<tr>
<th>Process Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear Session State</td>
<td>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.</td>
</tr>
<tr>
<td>See Also: &quot;Managing Session State Values&quot; on page 2-20 and &quot;Managing User Preferences&quot; in Oracle Application Express Administration Guide</td>
<td></td>
</tr>
<tr>
<td>Close Dialog</td>
<td>Closes the current modal or non-modal dialog page.</td>
</tr>
<tr>
<td>Data Manipulation</td>
<td>Data Manipulation process types are frequently used by wizards to support Data Manipulation Language (DML) actions. Application Builder supports the following declarative data manipulation processes: 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.</td>
</tr>
<tr>
<td>Form Pagination</td>
<td>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.</td>
</tr>
<tr>
<td>PL/SQL Code</td>
<td>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.</td>
</tr>
<tr>
<td>Plug-ins</td>
<td>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: &quot;Implementing Plug-ins&quot; on page 19-3.</td>
</tr>
<tr>
<td>Reset Pagination</td>
<td>Process to reset pagination of reports on the current page.</td>
</tr>
<tr>
<td>Send E-Mail</td>
<td>Process to send an email and optionally one or more attachments from the application.</td>
</tr>
<tr>
<td>User Preferences</td>
<td>Process to set User Preferences for the end user. See Also: &quot;Managing User Preferences&quot; in Oracle Application Express Administration Guide</td>
</tr>
<tr>
<td>Web Services</td>
<td>Implements a Web Service as a process on a page. Running the process submits a request to the service provider. See Also: &quot;How to Invoke a Web Service as a Process&quot; on page 25-20.</td>
</tr>
</tbody>
</table>
Process type - Select a process type as described in Table 17–4:

<table>
<thead>
<tr>
<th>Process Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear Session State</td>
<td>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: &quot;Managing Session State Values&quot; on page 2-20 and &quot;Managing User Preferences&quot; in Oracle Application Express Administration Guide</td>
</tr>
<tr>
<td>Close Dialog</td>
<td>Closes the current modal or non-modal dialog page.</td>
</tr>
<tr>
<td>Data Manipulation</td>
<td>Data Manipulation process types are frequently used by wizards to support Data Manipulation Language (DML) actions. Application Builder supports the following declarative data manipulation processes: 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.</td>
</tr>
<tr>
<td>Form Pagination</td>
<td>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.</td>
</tr>
<tr>
<td>PL/SQL Code</td>
<td>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.</td>
</tr>
<tr>
<td>Plug-ins</td>
<td>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: &quot;Implementing Plug-ins&quot; on page 19-3.</td>
</tr>
<tr>
<td>Reset Pagination</td>
<td>Process to reset pagination of reports on the current page.</td>
</tr>
<tr>
<td>Send E-Mail</td>
<td>Process to send an email and optionally one or more attachments from the application.</td>
</tr>
<tr>
<td>User Preferences</td>
<td>Process to set User Preferences for the end user. See Also: &quot;Managing User Preferences&quot; in Oracle Application Express Administration Guide</td>
</tr>
<tr>
<td>Web Services</td>
<td>Implements a Web Service as a process on a page. Running the process submits a request to the service provider. See Also: &quot;How to Invoke a Web Service as a Process&quot; on page 25-20.</td>
</tr>
</tbody>
</table>

4. Follow the on-screen instructions.

**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 Component View" on page 9-5.


3. Edit the appropriate attributes. For more information, see field-level Help and the sections that follow.

4. Click Apply Changes.

**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:


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.

**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:


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.
Making a Page Process Conditional

To create a conditional page process:


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 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.

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" on page 16-20

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?” on page 17-17.
About On Demand Application Processes

An On Demand process is a 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.

About Application Processes that Execute On New Instance

Typically an application process runs at the same point across multiple pages in an application. Processes having a Process Point of **On New Instance** are the exception. These types of processes are useful when you only need to retrieve information once within a user's session. For example, if the application items do not depend on the logged in user, you can initialize them using the Process Point **On New Instance**. To look up information that depends on the user, you can also use **After Authentication**.

If you configure applications to share the same session by setting the same cookie name in the authentication scheme, **On New Instance** and **After Login**, the application processes fire whenever Oracle Application Express first processes a request for the application (that is, On New Instance), or processes a request of an authenticated user for the application (that is, After Authentication).

About Running an On Demand Process from a Page Request

You can have a page request run an On Demand process by using the following syntax:

```
f?p=application_id:page_id:session:APPLICATION_PROCESS=process_id
```

Where:

- **application_id** is the application ID or alphanumeric alias
- **page_id** is the page number or alphanumeric alias
- **session** is the session ID
Understanding Application Processes

- **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 Application 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" on page 2-22

**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`.

**See Also:**  "Creating a Global Page to Display Components on Every Page" on page 14-3

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`.

**Creating an Application Process**

To create an application process:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.

2. Under Application Logic, select **Application Processes**.

   The Application Processes page appears.

3. Click **Create**.
4. For Identification:
   a. Name - Enter a name for the application process.
   b. Sequence - Specify the sequence number for this process. The sequence number determines the order in which the process is evaluated relative to other processes that execute at the same point.
   c. Point - Identify the point during the processing of each page in the application that this process would run. Note the option, On Demand, only runs when requested by a page process.
   d. Click Next.

5. For Source:
   a. PL/SQL code - Enter the PL/SQL code of your application process. The code will automatically be wrapped with a Begin and End;. 
   c. Click Next.

6. For Conditionality:
   a. Condition Type - Select a condition type that must be met in order for this process to execute.
   b. Expression 1 and Expression 2 - Use these attributes to conditionally control whether the process executes. Enter values in this attribute based on the specific condition type you select. The process executes if the specified condition is met.
   c. Click Create Process.

Editing Application Process Attributes
To edit an existing page process:
1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application 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. Click the process name.
4. Edit the appropriate attributes. For more information, see field-level Help.
5. Click Apply Changes.

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:


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**.

**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 **Application 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 “About the Actions Menu” on page 10-58.

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?" on page 17-17.

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:

```plsql
BEGIN
  :LAST_PAGE := nvl(:CURRENT_PAGE,:APP_PAGE_ID);
  :CURRENT_PAGE := :APP_PAGE_ID;
  RETURN :LAST_PAGE;
END;
```

About Application Computations that Execute On New Instance

Typically an application computation runs at the same point across multiple pages in an application. The exception is computations having a Computation Point of On New Instance. These types of computations are useful when you only need to retrieve information once within a user’s session. For example, if the application items do not depend on the logged in user, you can initialize them using the Computation Point On New Instance. To look up information that depends on the user, you can also use After Authentication.

If you configure applications to share the same session by setting the same cookie name in the authentication scheme, On New Instance and After Login, the application processes fire whenever Oracle Application Express first processes a
request for the application (that is, On New Instance), or processes a request of an authenticated user for the application (that is, After Authentication).

**Creating an Application Computation**

To create an application computation:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application Builder.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Shared Components page appears.

2. Under Logic, select Application Computations.

3. Click Create.

4. Edit the following attributes:
   a. Item, Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
   b. Item, Computation Item - Select the item this computation affects.
   c. Frequency, Computation Point - Select a process point at which this computation should be performed. Selecting After Submit causes the computation to be performed only after the page is displayed and then submitted.
   d. Computation, Computation Type - Select the manner in which this computation is performed.
   e. Computation, Computation - Enter the computation logic that corresponds to the computation type. See field-level Help for examples.
   f. Computation, Computation Error Message - Enter the error message that displays if the computation fails.
   g. Authorization, Authorization Scheme (optional) - Select an authorization scheme which must evaluate to True in order for this computation to execute.
   h. Condition, Condition Type (optional) - Select a condition type that must be met in order for this computation to execute.
   i. Build Option (optional) - Select a build option for this component. See "Using Build Options to Control Configuration" on page 24-31.

5. Click Create Computation.

**Accessing the Application Computation History Report**

Once you create an application computation, you can view the Application Computation History report. The Application Computation History report displays a history of recently changed application computations by date.

To access the Application Computation History report:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application Builder.
   b. Select an application.
c. On the Application home page, click **Shared Components**.
   
The Shared Components page appears.

2. Under Logic, select **Application Computations**.
3. Select the **History** at the top of the page.
4. You can customize the appearance the page using the Search bar at the top of the page. Available controls include:
   - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
   - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
   - **Go button** - Executes a search or applies a filter.
   - **Actions menu** - Use the Actions menu to customize the report view. See "About the Actions Menu" on page 10-58.

---

**Using the Attribute Dictionary**

This section explains how to use the Attribute Dictionary.

- **About the Attribute Dictionary**
- **Accessing the Manage Attribute Dictionary**

---

**About the Attribute Dictionary**

The Attribute Dictionary contains a set of attributes about a column that are used in creating forms and reports. The definitions are matched by column name and a particular definition can be shared among several columns by using synonyms.

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.

---

**Accessing the Manage Attribute Dictionary**

To access the Manage Attribute Dictionary page:

1. On the Workspace home page, click the **Application Builder** icon.
   
The Application 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**.
Managing Shared Components

Shared components can display or be applied on any page within an application. This section describes the advantages of using shared components and specifically how to manage static application files, static workspace files, images, application-level items, lists of values, lists, shortcuts, and report output.

- Working with Shared Components
- Managing Component Settings
- Managing Static Application Files
- Managing Static Workspace Files
- Managing Application-Level Items
- Creating Lists of Values at the Application Level
- Creating Lists
- Using Shortcuts
- Managing Report Output

See Also: "Application Builder Concepts" on page 2-1, "Creating Database Applications" on page 7-1, "Controlling Page Layout" on page 14-1, and "Adding Navigation" on page 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
- Accessing the Shared Components Page
- About the Shared Components Icon
- Shared Components Page

See Also: "Exporting an Application and Application Components" on page 24-12

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
Working with Shared Components

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" on page 18-1

### Accessing the Shared Components Page

To access the Shared Components page:

1. On the Workspace home page, click **Application 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.

### 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 Application Builder pages including the Application home page, Page Designer, Component View, Supporting Objects, and Utilities.

### Shared Components Page

This section describes the regions and links on the Shared Components page.

- **Application Logic**
- **Security**
- **Other Components**
- **Navigation**
- **User Interface**
- **Files**
- **Data References**
- **Reports**
- **Globalization**
- **Tasks Region**

**See Also:**  "Accessing the Shared Components Page" on page 18-2
Working with Shared Components

Application Logic

Table 18–1 describes the links under Application Logic on the Shared Components page.

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Definition Attributes</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Managing Application Attributes&quot; on page 7-13</td>
</tr>
<tr>
<td>Application Items</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Managing Application-Level Items&quot; on page 18-15</td>
</tr>
<tr>
<td>Application Processes</td>
<td>Use application processes to run PL/SQL logic:</td>
</tr>
<tr>
<td></td>
<td>■ At specific points for each page in an application</td>
</tr>
<tr>
<td></td>
<td>■ As defined by the conditions under which the process is set to execute</td>
</tr>
<tr>
<td></td>
<td>■ Upon the creation of a new session</td>
</tr>
<tr>
<td></td>
<td>Note that On Demand processes execute only when called from a page-level On Demand process or when called using Ajax from the browser.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Understanding Application Processes&quot; on page 17-25</td>
</tr>
<tr>
<td>Application Computations</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Understanding Application Computations&quot; on page 17-30</td>
</tr>
</tbody>
</table>

Security

Table 18–2 describes the links under Security on the Shared Components page.

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Attributes</td>
<td>Use the Edit Security Attributes page to configure general security attributes for all pages within an application.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Configuring Security Attributes&quot; on page 7-19</td>
</tr>
<tr>
<td>Authentication Schemes</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>See Also: &quot;Establishing User Identity Through Authentication&quot; on page 20-37</td>
</tr>
</tbody>
</table>
Table 18–2 (Cont.) Security Links on the Shared Components Page

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
</table>
| Authorization Schemes | Authorization restricts user access to specific controls or components based on predefined user privileges.  
  See Also: "Providing Security Through Authorization" on page 20-49         |
| Session State Protection | Session State Protection is a built-in functionality that prevents hackers from tampering with the URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy.  
  See Also: "Understanding Session State Protection" on page 20-21             |
| 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" on page 24-31.     |

Other Components

Table 18–3 describes the links under Security on the Shared Components page.

Table 18–3 Other Component Links on the Shared Components Page

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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" on page 18-18 |
| Plug-ins          | Application 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" on page 19-3                                  |
| 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" on page 18-7                           |
| 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" on page 18-34                                      |

Navigation

Table 18–4 describes the links under Navigation on the Shared Components page.
### Table 18–4 Navigation Links on the Shared Components Page

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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" on page 18-20                                                                                               |
| 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 *Universal Theme*.  
|                  | **See Also:** "About Navigation Menus" on page 18-21                                                                                       |
| 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" on page 16-1                                                                                                 |
| 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" on page 16-7                                                                                          |
| 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" on page 16-15                                                                              |

### User Interface

**Table 18–5** describes the links under User interface on the Shared Components page.

### Table 18–5 User Interface Links on the Shared Components Page

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
</table>
| User Interface Attributes | Use User Interface page to specify user interface options for an application.  
|                     | **See Also:** "Managing the Application User Interface" on page 7-29                                                                        |
| Themes              | A theme is a named collection of templates that defines the application user interface.                                                     
|                     | **See Also:** "Using Themes" on page 13-1                                                                                                  |
| 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" on page 13-23                                                                                       |

### Files

**Table 18–6** describes the links under User interface on the Shared Components page.
Table 18–6  Files Links on the Shared Components Page

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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" on page 18-8                                                                                       |
| 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" on page 18-12                                                                                      |

Table 18–7  Data References Links on the Shared Components Page

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
</table>
| Data Loading              | 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" on page 19-8                                                                        |
| Web Service References    | Web service references in Application 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" on page 25-7                                                                                      |

Table 18–8  Data References Links on the Shared Components Page

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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" on page 18-38                                                            |
| 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" on page 18-41                                               |

Table 18–9  describes the links under User interface on the Shared Components page.

Data References

Table 18–7 describes the links under User interface on the Shared Components page.

Reports

Table 18–8 describes the links under User interface on the Shared Components page.

Globalization

Table 18–9 describes the links under User interface on the Shared Components page.
Managing Component Settings

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. See "Exporting Application Components" on page 24-16.

- **Manage Supporting Objects** - Links to the Supporting Objects page. Use this page to create a packaged application. See "How to Create a Custom Packaged Application" on page 24-6.

- **Edit Application Comments** - Links to the Application Comments page. Use this page to enter comments specific to the currently selected application. See "Adding Database Application Comments" on page 7-58.


- **Developer Comments Calendar** - Links to Developer Comments Calendar page. See "Viewing the Developer Comments Calendar" on page 7-58.

Managing Component Settings
This section describes how to use the Component Settings page to configure application-level values for built-in Oracle Application Express components and installed plug-ins.

- **Editing Component Settings**

- **Viewing the Component Settings History Report**

Editing Component Settings
To edit component settings:

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Globalization Attributes</td>
<td>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: &quot;Configuring Globalization Attributes&quot; on page 7-26 and &quot;Understanding Application Translation and Globalization Support&quot; on page 21-1</td>
</tr>
<tr>
<td>Text Messages</td>
<td>Text messages are named text strings that can be called from the PL/SQL code you write. This PL/SQL can be anonymous blocks within page processes and page regions, or in packages and procedures. See Also: &quot;Translating Messages&quot; on page 21-13</td>
</tr>
<tr>
<td>Translate Application</td>
<td>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: &quot;Understanding Application Translation and Globalization Support&quot; on page 21-1 and &quot;Understanding the Translation Process&quot; on page 21-6</td>
</tr>
</tbody>
</table>
Managing Static Application Files

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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**.
   The Component Settings page displays as an interactive report. To customize the report, use the Search bar at the top of the page. To learn more, see "About the Search Bar" on page 10-57.

3. Edit the settings and click **Apply Changes**.

Viewing the Component Settings History Report

You can view a report of modifications made to Oracle Application Express components and installed plug-ins on the Component Settings History page.

To view the Component Settings History report:
1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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. To learn more, see "About the Search Bar" on page 10-57.

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
- Uploading Static Application Files
- Viewing the Static Application File Report
- About Referencing Static Application Files
- Downloading Static Application Files
- Deleting Static Application Files

See Also: "Managing Static Workspace Files" on page 18-12 and "About Managing Images" on page 14-25
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 Application 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://contentDevelivervNetwork.com/myFiles/

5. Click Apply Changes.

Uploading Static Application Files

To upload a static application file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application Builder.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Shared Components page appears.
2. Under Files, select Static Application Files.
   The Static Application Files page appears.
3. To upload a file, click Upload File.
4. On Upload:
   a. Directory - Enter the name of the directory where the file should be stored. If no directory is specified, the file is stored in the root directory.
   b. File - Enter the location of the file to be added to the application.
   c. File Character Set - Indicate the character set encoding for the file to be uploaded. This attribute only applies to text files.
   d. Unzip File - If Yes, the uploaded Zip file is automatically unzipped and stored in the specified directory. Note that the actual Zip file is not stored. This attribute is ignored if the uploaded file is not a Zip file.

5. Click Upload or Upload and Upload Another.
Viewing the Static Application File Report

To view details about an uploaded static application file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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" on page 10-57.

   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**.

About 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" on page 18-10, "About Using Substitution Strings" on page 2-25, "APP.Images" on page 2-30, and "Defining an Application Logo" on page 7-31

Downloading Static Application Files

This section describes how to download static application files.

- **Downloading a Static Application File**
- **Downloading All Static Application Files in a Zip**

**Downloading a Static Application File**

To download an uploaded static application file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
2. Under Files, select **Static Application Files**. The Static Application Files page appears.

3. If not already selected, click **View Report** icon. A report appears.

4. Locate the file to be downloaded and click the **Download** link.

**Downloading All Static Application Files in a Zip**
To download all static application files in a Zip:
1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**. The Shared Components page appears.

2. Under Files, select **Static Application Files**. The Static Application Files page appears.

3. Click **Download as Zip**.

**Deleting Static Application Files**
This section describes how to delete uploaded static application files.

- **Deleting a Static Application File**
- **Deleting All Static Application Files**

**Deleting a Static Application File**
To delete an uploaded static application file:
1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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. To learn more, see "About the Search Bar" on page 10-57.

4. Select the file to be deleted.

5. On the Edit page, click **Delete**.

**Deleting All Static Application Files**
To delete all uploaded static application files:

1. Navigate to the Shared Components page:
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
- Viewing Static Workspace File Report
- About Referencing Static Workspace Files
- Downloading Static Workspace Files
- Deleting Static Workspace Files

See Also: "Managing Static Application Files" on page 18-8 and "About Managing Images" on page 14-25

Uploading Static Workspace Files

To upload a static workspace file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
      The Shared Components page appears.

2. Under Files, select **Static Application Files**.
   The Static Application Files page appears.

3. To upload a file, click **Upload File**.

4. On Upload:
   a. Directory - Enter the name of the directory where the file should be stored. If no directory is specified, the file is stored in the root directory.
   b. File - Enter the location of the file to be added to the application.
   c. File Character Set - Indicate the character set encoding for the file to be uploaded. This attribute only applies to text files.
   d. Unzip File - If yes, the uploaded Zip file is automatically unzipped and stored in the specified directory. Note that the actual Zip file is not stored. This attribute is ignored if the uploaded file is not a Zip file

5. Click **Upload** or **Upload and Upload Another**.
Viewing 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 Application 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. To learn more, see "About the Search Bar" on page 10-57.
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.

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" on page 18-13, "Using Built-in Substitution Strings" on page 2-26, "WORKSPACE.IMAGES" on page 2-41, and "Defining an Application Logo" on page 7-31

Downloading Static Workspace Files

This section describes how to download static workspace files.

- Downloading a Static Application File
- Downloading All Static Application Files in a Zip

Downloading a Static Workspace File

To download an uploaded static workspace file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application Builder.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Shared Components page appears.

2. Under Files, select Static Workspace Files.
   The Static Workspace Files page appears.
3. If not already selected, click **View Report** icon.
   A report appears.
4. Locate the file to be downloaded and click the **Download** link.

### Downloading All Static Workspace Files in a Zip

To download all static workspace files in a Zip:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
      The Shared Components page appears.
2. Under Files, select **Static Workspace Files**.
   The Static Application Files page appears.
3. Click **Download as Zip**.

### Deleting Static Workspace Files

This section describes how to delete uploaded static workspace files.

- Deleting a Static Workspace File
- Deleting All Static Workspace Files

#### Deleting a Static Workspace File

To delete an uploaded static workspace file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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. To learn more, see "About the Search Bar" on page 10-57.
4. Select the file to be deleted.
5. On the Edit page, click **Delete**.

#### 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 **Application Builder**.
   b. Select an application.
c. On the Application home page, click **Shared Components**.
The Shared Components page appears.

2. Under Files, select **Static Workspace Files**.
The Static Workspace Files page appears.

3. Click **Delete All Files**.

Managing Application-Level Items

Application-level items do not display, but are used as global variables to the application. You can use an application item as a global variable.

- Creating an Application-Level Item
- Viewing Application Item Usage
- Editing Application-Level Item Attributes
- About Creating an Application-Level Item that Functions as a Global Variable

See Also: "About the Differences Between Page Items and Application Items" on page 15-1 and "Referencing Item Values" on page 15-2

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 **Application 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. To learn more, see "About Creating an Application-Level Item that Functions as a Global Variable" on page 18-17.
      - 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:
Managing Application-Level Items

- **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.

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 Application Builder.
   b. Select an application.
c. On the Application home page, click **Shared Components**.
   The Shared Components page appears.

2. Under Application Logic, select **Application Items**.
   The Application Items page appears.

3. Click the **Utilization** tab at the top of the page.

**Editing Application-Level Item Attributes**

Once you create an application-level item, you can edit it on the **Create/Edit Application Item** page.

To edit application-level item attributes:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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. See "Viewing Field-Level Help" on page 1-21.

5. Click **Apply Changes**.

   See Also: "Understanding Cross-Site Scripting Protection" on page 20-16

**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" on page 18-15

Creating Lists of Values at the Application Level

This section describes how to create lists of values.

- What is a List of Values?
- Creating a Named LOV at the Application Level
- About Static LOVs
- Editing an Existing LOV
- About Referencing Session State Within an LOV
- Accessing LOV Reports

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" on page 15-8 and "Working with Shared Components" on page 18-1

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 Application Builder.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Shared Components page appears.

2. Under Other Components, select List of Values.
   The Lists of Values page appears.
3. To create a LOV, click **Create**.
4. Follow the on-screen instructions.

New named LOVs are added to the List of Values repository. To add the LOV to a page, you must edit the appropriate item or column and select the named LOV.

**About Static LOVs**

Static LOVs are based on a static list of display values and return values you specify when you run the Create LOV Wizard. To create a static LOV, run the Create LOV Wizard and select the LOV type **Static**. Oracle Application Express stores the display values, return values, and sort sequence you specify in the List of Values repository. Once you add a static LOV to the repository, you can create an item and display it as a check box, radio group, select list, or popup list based on this definition.

**Editing an Existing LOV**

To edit an existing LOV, select the LOV on the Lists of Values page.

- **Editing a LOV**
- **Editing LOVs in Bulk**

**Editing a LOV**

To edit a LOV:

1. Navigate to the Shared Components page:
   - a. On the Workspace home page, click **Application 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. See "Viewing Field-Level Help" on page 1-21.

5. Click **Apply Changes**.

**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 **Application Builder**.
   - b. Select an application.
   - c. On the Application home page, click **Shared Components**.

Managing Shared Components 18-19
Creating Lists

The Shared Components page appears.

2. Under Other Components, select **List of Values**.
   By default, LOVs display as icons.
3. Click the **Grid Edit** tab.
4. Edit the display text and click **Apply Changes**.

**About Referencing Session State Within an LOV**

You can reference session state by using bind variables. Keep in mind that referencing session state makes an LOV a bit less reusable, but is still a recommended development practice. In the following example, this LOV only works if the item called `my_deptno` contains a valid department number.

```
SELECT ename, empno FROM emp WHERE deptno = :P1_DEPTNO
```

**Accessing LOV Reports**

To access LOV reports:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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.

**Creating Lists**

This section explains how to add shared collection of links (or 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**
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.


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.
Creating Lists

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 at 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.

See Also: "Managing the Application User Interface" on page 7-29 and "Editing Themes" on page 13-8

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 application uses two levels of tabs, you must convert it to one level before changing the theme.

Creating Static Lists
This section explains about to create a static list.

- About the Process of Creating a Static List
- Creating a Static List from Scratch
- About Adding Entries and Sublists to a Static List
- Adding Entries or Sublists from Scratch
- Copying Static List Entries Between Lists
- Reparenting Static List Entries
- Managing Orphaned Static List Entries
- Resequencing Static List Entries
About the Process of Creating a Static List
The process of creating a static list involves the following steps:

■ **Step 1:** Create the list by running the Create List Wizard.

■ **Step 2:** Specify whether to create the list from scratch or by copying an existing list.

  You can add list entries by either creating them from scratch or by copying entries from an existing list. If you copy an entire list, you also copy all of its list entries.

■ **Step 3:** If creating a list from scratch, you are prompted to select a list type. Select Static.

  Each list element has a display condition, which enables you to control when it displays. You can define a list element to be either current or non-current for a specific page.

■ **Step 4:** Add the list to a page by creating a List region.

How you perform these steps is a personal preference. You can perform the first three steps when you run the Create List Wizard. Alternatively, you can perform these steps one at a time.

  **See Also:** "Adding a List to a Page" on page 18-32, "Editing List Attributes" on page 18-32, and "Accessing List Reports" on page 18-33

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 **Application 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.

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:
Creating Lists

- Do not create list region(s)
- Create list region on current page
- Create list region for each target page

b. Region Position - If creating a new region, select the region position.
c. Region Template - If creating a new region, select a region template.
d. List Template - Select the list template to use to control the appearance of your list.
e. Click Create List.

About Adding Entries and Sublists to a Static List

Once you create a list, you must add entries to it. You can add list entries when creating a list from scratch, copying a list entry from within a list, or copying existing entries from one list to another.

You can also create hierarchical lists that contain sublists. To create a hierarchical list, you must:

- Select a list template that supports hierarchical lists. To determine which list templates support hierarchical lists, look for templates having the naming convention “with Sublist.”
- Select a Parent List Item when you create each list entry.

See Also: "Creating Custom Themes" on page 13-23 and "List Templates" on page 13-40

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 Application Builder icon.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Lists page appears.
2. Select a list.
   The List Details page appears.
3. Click Create List 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 - Identify the file name for the image used to display this list entry. List templates control this attribute.
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:

```html
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:

```html
...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:

```html
http://www.yahoo.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. "Editing Themes" on page 13-8 and "Optimizing a Page for Printing" on page 14-2.

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 - Specify the page numbers on which to clear cache.

To learn more about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

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:
Creating Lists

1. 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.

2. 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 about an attribute, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

7. Under Conditions, specify the appropriate information to make the list entry conditional.


   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, such as www.google.com, you can count clicks. For more information, see "COUNT_CLICK Procedure" in Oracle Application Express API Reference.

11. To specify additional attributes, use the User Defined Attributes section. 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.

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 Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
   e. Under Navigation, click Lists.

      The Lists page appears.

2. Select the list to copy.

3. On the Tasks list, click Copy List Entries from one List to Another.

4. From Copy List Entries, select the list to copy to and click Next.
5. Click **Copy List Entries**.

**Reparenting Static List Entries**

Use the Reparent List Entries page to manage list entry hierarchy.

To reparent list entries:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the **Application Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Lists**.
      The Lists page appears.

2. Select the list.

3. From the Tasks list, click **Reparent List Entries within this List**.

4. Use Start With to restrict your view to a subset of hierarchy. Make a selection and click **Go**.

5. From Reparent To, select a new parent.

6. Select the entries to move.

7. Click **Reparent Checked Entries**.

**Managing Orphaned Static List Entries**

An orphaned list entry is a list entry which has a parent, but the parent is no longer a member of the current list. When a list entry becomes orphaned, remove the parent entry.

To manage orphaned list entries:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the **Application Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Lists**.
      The Lists page appears.

2. Select the list.

3. From the Tasks list, click **Manage Orphaned List Entries**.

4. Select the entries to remove.

5. Click **Clear Parent Entries**.

**Resequencing Static List Entries**

You can resequence list entries in increments of 10 on the Clean Up page.

To clean up list entries:
1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
   e. Under Navigation, click Lists.
      The Lists page appears.
2. Select the list.
3. From the Tasks list, click Clean Up List Entries.
4. Click Clean Up.

Creating Dynamic Lists
This section explains about to create a dynamic list.

- About the Process of Creating a Dynamic List
- Understanding Dynamic List Syntax
- Creating a Dynamic List from Scratch

About the Process of Creating a Dynamic List
A dynamic list enables you to 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" on page 18-32, "Editing List Attributes" on page 18-32, and "Accessing List Reports" on page 18-33

Understanding Dynamic List Syntax
You create a dynamic list using a SQL query or a function returning a SQL Query.

Syntax for a SQL query:
```
SELECT level, labelValue label,  
[targetValue] target,  
[is_current] is_current_list_entry,  
[imageValue] image,  
[imageAttributeValue] image_attribute,  
[imageAltValue] image_alt_attribute,  
[attribute1] attribute1,  
[attribute2] attribute2,
```
[attribute3] attribute3,
[attribute4] attribute4,
[attribute5] attribute5,
[attribute6] attribute6,
[attribute7] attribute7,
[attribute8] attribute8,
[attribute9] attribute9,
[attribute10] attribute10

FROM ...
WHERE ...
ORDER BY ...

Syntax for a function returning a SQL query:

RETURN
'SELECT level, labelValue label,'||
  '[targetValue] target,'||
  '[is_current] is_current_list_entry,'||
  '[imageValue] image,'||
  '[imageAttributeValue] image_attribute,'||
  '[imageAltValue] image_alt_attribute,'||
  '[attribute1] attribute1,'||
  '[attribute2] attribute2,'||
  '[attribute3] attribute3,'||
  '[attribute4] attribute4,'||
  '[attribute5] attribute5,'||
  '[attribute6] attribute6,'||
  '[attribute7] attribute7,'||
  '[attribute8] attribute8,'||
  '[attribute9] attribute9,'||
  '[attribute10] attribute10'|
'FROM ...'||
'WHERE ...'||
'ORDER BY ...';

Where:

- **level** and **labelValue** are required.
- **level** - For hierarchical lists, the level parameter should be supplied. For non-hierarchical lists, this parameter can be set to NULL.
- **labelValue** - Text to appear as list entry.
- **targetValue** - Target URL to branch to when list entry is selected.
- **is_current** - Controls the behavior of the list entry. Values include:
  - **NULL** - Currency of target is based upon Target URL.
  - 'YES' - List entry is always current.
  - 'NO' - List entry is not current.
- **imageValue** - The name of image to be display on the list entry
- **imageAttributeValue** - Attributes of the image, such as the width or height
- **imageAltValue** - Value for Image ALT tag, required for accessibility purposes in templates where the user must click the image.
- **attribute1** to **10**: These attributes tie in with the existing ten User Attributes exposed on the Static List Entry page.
Creating Lists

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 Application Builder icon.
   b. Select an application.
   c. On the Application home page, click Shared Components.
   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.
   c. 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.
   d. 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.

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
- Copying a List Between Applications
Copying a List from the Current Application
You can copy list entries from one list to another.

To copy list entries between lists:
1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the Application Builder icon.
   c. Select an application.
   d. On the Application home page, click Shared Components.
   e. Under Navigation, click Lists.
      The Lists page appears.
2. Select the list to copy.
3. On the Tasks list, click Copy List.
4. On Copy List, select List in this application and click Next.
5. For Identify List:
   a. Copy List - Select the list to be copied.
   b. New List Name - Enter a new name for the copied list.
   c. Click Copy.

Copying a List Between Applications
You can copy a list from the current application or from another application.

To copy a list:
1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the Application 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 new name for the new list.
   d. Click Copy.
Adding a List to a Page

Once you create a list and list entries, the next step is to add it to a page by creating a region and specifying the region type as List.

See Also: "Creating a Region in Page Designer" on page 14-4

To add a list to a page:

1. Navigate to the appropriate page. See "Viewing a Page in Component View".
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.
   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.
   d. Display Point - Identify a display point for this region.
   e. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
   f. Click Next.
5. For Source:
   a. List - Select the list you want to add.
   b. List Template - Choose a template to control the look of the list.
   c. Click Next.
6. 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" on page 2-12.
   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" on page 20-49.
   c. Click Create Region.

Repeat these procedures for each page where you would like to add a list.

See Also: "Creating a New Template" on page 13-26 and "List Templates" on page 13-40 for information about altering list display

Editing List Attributes

Once you create a list, you can edit it on the Lists page.

To edit a list:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
b. Click the **Application Builder** icon.

c. Select an application.

d. On the Application home page, click **Shared Components**.

e. Under Navigation, click **Lists**.

The Lists page appears.

2. Select a list.

The List Details page appears.

3. To change the list name or build options, click the **Edit List** button.

4. To change the list name or build options:

   a. Click the **Edit List** button.

   b. Edit the appropriate attributes.

   c. Click **Apply Changes**.

5. To edit the list sequence, link text, target, or current status for all list entries, click **Grid Edit**.

   a. To edit the list sequence, link text, target, or current status for all list entries, click **Grid Edit**.

   b. Edit the appropriate attributes.

   c. Click **Apply Changes**.

**Accessing List Reports**

You can view the Unused, Conditional Entries, Utilization, and History reports by clicking the appropriate tab at the top of the Lists page. Note that these reports only display after you create a list.

To view list reports:

1. Navigate to the Lists page.

   a. Navigate to the Workspace home page.

   b. Click the **Application 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.
Using Shortcuts

This section describes how to create and use shortcuts.

- What is a Shortcut?
- Understanding Shortcut Types
- Defining and Referencing Shortcuts
- Editing Existing Shortcuts
- Accessing Shortcut Reports

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" on page 14-4.
- Region Header and Footer Text attribute. See "Specifying a Region Header and Footer" on page 14-6.
- Item Label attributes, Pre Element Text, Post Element Text, and Default Value attribute.
- Region Templates attributes. See "Creating Custom Themes" on page 13-23.

Understanding Shortcut Types

This section describes the types of shortcut supported by Oracle Application Express.

- Supported Shortcut Types
- Text with JavaScript Escaped Single Quotes
- Message
- Message with JavaScript Escaped Single Quotes

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
Using Shortcuts

Managing Shared Components

- HTML Text with Escaped Special Characters
- Image
- Text with JavaScript Escaped Single Quotes
- Message
- Message with JavaScript Escaped Special Quotes

**Text with JavaScript Escaped Single Quotes**
Use this type of shortcut to reference a shortcut inside of a JavaScript literal string. This shortcut defines a text string. When the shortcut is referenced, it escapes the single quotation marks required for JavaScript.

**Message**
Use this type of shortcut to reference a translatable message at runtime. Since this shortcut does not have a shortcut body, the name of the shortcut must match the corresponding message name. At runtime, the name of the shortcut expands to the text of the translatable message for the current language.

**Message with JavaScript Escaped Single Quotes**
Use this type of shortcut to reference a shortcut inside of JavaScript literal string and reference a translatable message at runtime.

See Also: "Understanding Application Translation and Globalization Support" on page 21-1

**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
- Copying an Existing Shortcuts
- Referencing a Shortcut

**Defining a Shortcut from Scratch**
To define a shortcut:
1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application 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.
Using Shortcuts

b. Type - Select a shortcut type. Options include:
   - PL/SQL Function Body
   - HTML Text
   - HTML Text with Escaped Special Characters
   - Image
   - Text with JavaScript Escaped Single Quotes
   - Message
   - Message with JavaScript Escaped Special Quotes

c. Shortcut - Enter the text of the shortcut. The shortcut must be less than 30,000 characters in length.

d. Error Text - Enter the text to display if an error should occur during the processing of a shortcut.

e. Build Option - Select a build option for this component.

6. Click Create.

New shortcuts are added to the Shortcuts page (also referred to as the Shortcut repository).

Copying an Existing Shortcuts

To copy an existing shortcut:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application 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:

7. 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).

Referencing a Shortcut

You reference shortcuts using the following syntax:

"MY_SHORTCUT"
The shortcut name must be capitalized and enclosed in quotation marks.

**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 *Application 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. See "Viewing Field-Level Help" on page 1-21.
5. Click *Apply Changes*.

**Accessing Shortcut Reports**

To access shortcut reports:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click *Application Builder*.
   b. Select an application.
   c. On the Application home page, click *Shared Components*.
      The Shared Components page appears.
2. Under Other Components, select *Shortcuts*.
3. Click one of the following tabs:
   - **Subscription** - Displays the Subscribed Shortcuts page which displays all subscribed shortcuts in your application.
   - **History** - Display the Shortcut History page which displays a history of recently changed shortcuts by date.

**Note:** The Subscription and History tabs only appear after you create a shortcut.

**Managing 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

This section explains how to print a report region by defining a report query.

- **About Report Queries**
- **Creating a Report Query**
- **Editing a Report Query**
- **Copying a Report Query**

#### About Report Queries

You can print a report region by defining a report query as a Shared Component. A report query identifies the data to be extracted. Unlike SQL statements contained in regions, report queries contain SQL statements that are validated when you save the query. Note that report queries must be SQL statements, not functions returning SQL statements.

You can associate a report query with a report layout and download it as a formatted document. If no report layout is selected, a generic layout is used. The generic layout is intended to be used to test and verify a report query. When using the generic layout option and multiple source queries are defined, only the first result set is included in the print document. The reports can include session state of the current application.

To make these reports available to end users, you then integrate them with an application. For example, you can associate a report query with a button, list item, branch, or other navigational component that enables you to use URLs as targets. Selecting that item then initiates the printing process.

#### Creating a Report Query

Note that the availability of the report query options depends on how your service administrator configures report printing for your instance. All options described in these steps may not be available to you.

To create a report query:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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.
– Excel - Microsoft Excel format. Note that this is not a true .xls file because the content is HTML-based.
– HTML
– XML - Extensible Markup Language.

c. Item - Select the item to hold the format value. You can use this item to determine the output format at runtime.

d. View File As - Select how the web browser opens the report query document. Options include:
   – Attachment - Displays a File Open/Save dialog box when the user clicks the report query download button.
   – Inline - Displays the report query document inside the Browser window.

e. Session State - Select this check box to reference session state values, such as the values of page items in the XML structure of your report.

f. Click Next.

5. For SQL Query, enter a SQL statement directly or click Query Builder to build a SQL statement by clicking and pointing.

To reference applications and page items in the SQL statement, reference them as bind variables. Click Set Bind Variables to be able to enter in values for each bind variable and test the query to ensure the expected output is returned.

6. For Download Definition, specify the following:

a. Data Source for Report Layout - Use this option to download a report definition export file.

   Tip: The report query definition is what a developer would use as the source for tools like BI Publisher Desktop or Altova Stylevision to create the report layout. Some tools accept both an XML export of the data as well as an XML scheme definitions as source, other tools may only accept one or the other. If the source query does not return data at design time, then XML schema provides a way to export the definition without data.

Select XML Data or XML Schema and click Download.

XML Schema creates XSD file (that is, a XML schema definition of your XML structure.)

b. Add Query - Click this button to add another query to the report query. Adding another query enables the production of complex reports with more than one output component (for example, a chart and a report based on two different queries).

c. Create Report Query - Click this button to exit the wizard early. The Report Layout Source defaults to use the generic report layout.

d. Click Next.

7. For Upload Report Layout, specify the following:

a. Report Layout Source - Specify how the report layout is derived. If you choose the report based file layout then the layout name and report layout file must be provided.
b. Layout Name - Enter a layout name (optional). The default is the report query name.


d. Click Next.

8. For Confirm:

a. Query Name - Identifies the query.

b. Report Layout - Identifies the report layout you selected.

c. Output Format - Identifies the format for this report query.

d. Derive from Item - Identifies the item to hold the format information.

e. URL - To integrate this report with your application, use the displayed URL as the target for a button, list item, link, or other navigational component. End users can click a button, for example, to start the printing process.

f. Test Report - Click this to preview your report. If you have chosen to use a generic report layout then the Test Report does not produce an output.

g. Click Finish.

The Report Query is created and saved to Shared Components.

See Also: "About Report Printing Configuration Options" on page 10-84

Editing a Report Query
To edit a report query:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application Builder.

2. Under Reports, click Report Queries.

3. Select the appropriate report query.

4. On the Edit page, edit the appropriate attributes.

5. Click Apply Changes.

Copying a Report Query
To copy a report query:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application Builder.

2. Under Reports, click Report Queries.

4. On the Copy Report Query, select the query you want to copy, enter a name for the report query, and click **Copy**. The copy appears in the query list.

**Formatting a Report Region or Report Query Using Report Layouts**

This section describes how to use a report layout to format a report region or report layout.

- **About Report Layouts**
- **About Report Layout Options**
- **Creating a Report Layout**
- **Editing a Report Layout**
- **Copying a Report Layout**

**About Report Layouts**

To format either a classic report region or report query, you associate it with a report layout. Using report layouts renders the data in a printer-friendly format. If you do not select a report layout, a default XSL-FO layout is used. The default XSL-FO layout is always used for rendering Interactive Report regions.

When creating and using report layouts, you can:

- Take advantage of the default layouts for report regions and generic layouts for report queries provided with Oracle Application Express.
- Utilize the built-in XSL-FO-based layouts for report regions by copying and customizing the code. You can edit several attributes for report regions that control page size, fonts, colors, and so on.
- Create RTF or XSL-FO report layouts to customize the report look and feel. To use RTF report layouts, your Oracle Application Express service administrator must select the Advanced setting for your site. See "Configuring Report Printing" in *Oracle Application Express Administration Guide*.

**Note:** Interactive reports do not support a custom report layout.

**About Report Layout Options**

You can create a report layout based on one of these options:

- **Generic Columns** - A generic report layout works with most query result sets. With this layout, the number of columns is automatically adjusted when generating the printable document.

Many report layout attributes can be defined declaratively for report regions using the built-in XSL-FO default layout. This step allows for creating customizable copies of the built-in default XSL-FO layout, if additional control over the report layout is needed.

- **Named Columns** - A named column report layout is a query-specific report layout designed to work with a defined list of columns in the query result set. This type of layout is used for custom-designed layouts when precise control of the positioning of page items and query columns is required.
Note that the availability of the Report Layout options depends on how your site administrator configured the report printing settings at your site. All options described in these steps may not be available to you.

**See Also:** "Configuring Report Printing" in *Oracle Application Express Administration Guide*.

### Creating a Report Layout

To create a report layout:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
      
The Shared Components page appears.

2. Under Reports, click **Report Layouts**.

3. Click **Create**.

   The Create Report Layout wizard appears.

4. For Report Layout Type, select an option and click **Next**:
   - **Generic Columns (XSL-FO)** - Uses the default template to populate the report. In the next step, you can customize the default code.
   - **Named Columns (XSL-FO)** - Requires that you upload an XSL-FO or RTF file in the next step.

5. For Layout Source, review and edit the appropriate information. The options that appear on this page depend on the layout type you select:
   - If you selected **Generic Columns**:
     a. **Report Layout Name** - Enter a name to identify the report layout when associating it with a report query or report region.
     b. **Report Layout** - The report layout is the XSL-FO based definition of the page formatting. All attributes defining page size, orientation, fonts, styles, and so on, are defined in this section.
     c. **Report Column Heading** - Defines the look of each cell in the report heading row.
     d. **Report Column** - Defines the look of each cell for all report rows.
     e. **Report Column Width** - This width is computed at runtime or can be derived from the report column definition of a report region.
   - If you selected Named Columns:
     a. **Layout Name** - Enter a name to identify the report layout when associating it with a report query or report region.
     b. **Report Layout File** - Upload the file containing the report layout.

6. Click **Create Layout**.
Editing a Report Layout
You can edit a generic column report layout directly in Application Express. However, to edit a named column report layout, you must download the current file, edit it, and then upload it again.

To edit a report layout:
1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application Builder.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Shared Components page appears.
2. Under Reports, click Report Layouts.
3. On the Report Layouts page, select the layout you want to edit.
4. For generic column layouts, edit the layout directly on the Edit Report Layout page and click Apply Changes.
5. For named column layouts, click Download and save the file to your computer.
   Edit the file and then upload the updated version as a new report layout.

Copying a Report Layout
You can copy a report layout to edit and save.

To copy a report layout:
1. Navigate to the Shared Components page:
   a. On the Workspace home page, click Application 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.
4. On the Copy Report Layout page, select the layout you want to copy, enter a name for the copy, and click Copy.
   The copy appears in the layout list.
5. On Copy Report Layout:
   a. Copy Report Layout - Select the layout you want to copy.
   b. Report Layout Name - Enter a name for the copy.
   c. Click Copy.
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.

- Sending Email from an Application
- Implementing Plug-ins
- Understanding jQuery and jQuery UI Support
- Manually Refreshing Oracle Application Express Components

See Also: Oracle Application Express API Reference and "Deploying an Application" on page 24-1

Sending Email from an Application

This section describes how to configure and manage email in Oracle Application Express.

- About Configuring Email
- About Email Provisioning
- About the Mail Queue
- About the APEX_MAIL Package

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.

See Also: "Configuring Email" in Oracle Application Express Administration Guide

- About Enabling Network Services
- About Configuring Email in a Full Development Environment

About Enabling Network Services

If you are running Oracle Application Express with Oracle Database 11g or later, you must enable outbound mail. Starting with Oracle Database 11g Release 1 (11.1), the ability to interact with network services is disabled by default.
By default, the ability to interact with network services is disabled in Oracle Database 11g or later. Therefore, if you are running Oracle Application Express with Oracle Database 11g or later, you must use the `DBMS_NETWORK_ACL_ADMIN` package to grant connect privileges to any host for the `APEX_050000` 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" for your configuration scenario in *Oracle Application Express Installation Guide* and "Configuring Email" in *Oracle Application Express Administration Guide*

### About Configuring Email in a Full Development Environment

To configure Oracle Application Express to send mail in a full development environment your Instance administrator must log in to Oracle Application Express Administration Services, navigate to the Instance Settings page and configure Email attributes.

**See Also:** "Configuring Email" in *Oracle Application Express Administration Guide*

### About Email Provisioning

You can configure Oracle Application Express to automatically email users their login credentials when a new workspace request has been approved. To accomplish this, you must complete the email configuration process and select the provisioning status, Request with Email Verification.

**See Also:** "Configuring Email," "About Specifying How Workspaces Are Created" and "Selecting a Provisioning Mode" in *Oracle Application Express Administration Guide*

### About the Mail Queue

Instance administrators can manage email sent from applications by monitoring email messages in the mail queue and mail log.

**See Also:** "Managing the Mail Queue" in *Oracle Application Express Administration Guide*

### About the APEX_MAIL Package

You can send email from an Oracle Application Express application using the `APEX_MAIL` package. 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* and "Configuring Email" in *Oracle Application Express Administration Guide*
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
- Viewing Plug-in Examples
- Accessing the Plug-ins Page
- Creating a Plug-in
- Editing a Plug-in
- Adding Custom Attributes to a Plug-in
- Uploading Files Associated with a Plug-in
- Adding Events to a Plug-in
- Deleting a Plug-in
- Viewing the Plug-in Repository
- Importing a Plug-in from the Plug-in Page
- Exporting a Plug-in from the Plug-in Page
- Resetting the Plug-in Interactive Report
- Viewing Plug-in Utilization Page
- Viewing Plug-in History

See Also: "Viewing Plug-in Examples" on page 19-3 and "Viewing the Plug-in Repository" on page 19-8

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.

Viewing Plug-in Examples

To view plug-in implementation examples:
- Go to the Oracle Plug-in Repository.

See "Viewing the Plug-in Repository" on page 19-8.
Implementing Plug-ins

- Install the packaged application, *Sample Database Application*.
  See "Managing Packaged Applications in a Full Development Environment" on page 3-2 and "Installing and Running Sample Database Application" on page 3-6.
- Go to the Oracle Learning Library at the following location, click the link at the bottom of the page to access the library, then enter search criteria for Oracle Application Express (APEX) Product OBEs:
  
  http://www.oracle.com/technetwork/tutorials/index.html

**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 Application 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. To learn more, see "Customizing Interactive Reports as an End User" on page 10-57.

**Creating a Plug-in**

To create a plug-in:

<table>
<thead>
<tr>
<th>Tip:</th>
<th>To learn more about an attributes described in this section see field-level Help. See &quot;Viewing Field-level Help in Oracle Application Express&quot; on page 1-21.</th>
</tr>
</thead>
</table>

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.

| Tip: | To insure the internal name is a globally unique name worldwide, Oracle recommends that your organization domain name be used as a prefix to internal plug-in names. For example, a domain name of mycompany.com prefixed to a plug-in named Slider, would result in an internal name of COM.MYCOMPANY.SLIDER. |
c. Type (Required) - Select the type of component that can use this plug-in. Depending upon the plug-in type you select, the options under Callbacks and Standard Attributes differ. To learn more, see field-level Help.

d. Category - Only displays if the selected type is Dynamic Action. Select the category the plug-in is displayed under on the user interface.

5. Under Subscription:
   - Reference Master Plug-in From - To base this plug-in on another plug-in in this workspace, select the plug-in from the list. Otherwise, leave the field blank to make this the master copy of this plug-in.

6. Under Source:
   a. PL/SQL Code - Enter a PL/SQL anonymous block of code that contains the procedures for rendering, validating, executing, and performing Ajax callbacks for this plug-in. For performance reasons you can also store this code in a PL/SQL package in the database.
   b. Do not validate PL/SQL code (parse PL/SQL code at runtime only) - Select this option to parse the PL/SQL code at runtime only. Otherwise, the code is parsed when the plug-in is created.

7. Under Callbacks, configure that appropriate attributes. The attributes that display depend upon the plug-in type. To learn more about an attribute and view examples, see field-level Help.

   Tip: All Callback function names can reference a function of the anonymous PL/SQL code block, a function within a package or a standalone function in the database.

8. Under User Interfaces, select the display devices the Application Builder must support for this plug-in. Options include:
   - Desktop
   - Mobile

9. For Standard Attributes, select the attributes that apply to this plug-in. Standard Attributes do not display for some plug-ins. To learn more, see field-level Help.

10. Under Information:
    a. Version - Enter a string to identify the plug-in version.
    b. About URL - Enter a URL to the plug-in authors home page or to additional information about the plug-in.

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 "Adding Custom Attributes to a Plug-in" on page 19-6, "Uploading Files Associated with a Plug-in" on page 19-7, and "Adding Events to a Plug-in" on page 19-8.
Editing a Plug-in

To edit a plug-in:

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: "Creating a Plug-in" on page 19-4, "Adding Custom Attributes to a Plug-in" on page 19-6, "Uploading Files Associated with a Plug-in" on page 19-7, and "Adding Events to a Plug-in" on page 19-8

Adding Custom Attributes to a Plug-in

A plug-in attribute is used to prompt the developer for additional data in the Application Builder when the plug-in is used.

To add custom attributes to the plug-in:

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.
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
- Automatically Loading CSS and JavaScript Files

Uploading a File to Associate with a Plug-in

To upload a file:

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**.

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:

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**.
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:

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.

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:

2. Click the plug-in you want to delete.
   
   The Plug-in Create/Edit page appears.
3. Click Delete.
4. To confirm, click OK.

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:

2. Click View Plug-in Repository.
   
   The Oracle Application Express Plug-in Repository displays.

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 Application Builder home page. See "Importing Plug-ins" on page 24-25.

To import a plug-in:

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**.

### 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. See "Exporting Plug-ins" on page 24-20.

To export a plug-in from the Plug-in 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.

### 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:

2. On the Plug-ins page, click **Reset**.

### Viewing Plug-in Utilization Page

The Plug-in Utilization page displays which pages, components, and regions use each plug-in.
To view plug-in utilization:


2. Click **Utilization**.

   The Utilization page appears.

**Viewing Plug-in History**

The Plug-in History page shows the actions taken on each plug-in, the developer that performed the action and the date of each action.

To view plug-in history:


2. Click **History**.

   The History page appears.

**Understanding jQuery and jQuery UI Support**

Oracle Application Express includes the jQuery 1.11.2 and 2.1.3, jQuery UI 1.10.4, and jQuery Mobile - 1.4.4 JavaScript libraries. This section discusses the features available in jQuery UI and best practices when referencing jQuery libraries in your JavaScript code.

- About Updated JavaScript Libraries for jQuery and jQueryUI
- About Available jQuery UI Features
- Referencing the jQuery Library in Your JavaScript Code

**About Updated JavaScript Libraries for jQuery and jQueryUI**

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 2.1.3.
- 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://jqueryui.com/docs/Changelog](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:
Understanding jQuery and jQuery UI Support

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 suggestions 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.
Extending Application Capabilities

19-11


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 Application Builder for the following libraries/jquery-ui/ and replace it with #JQUERYUI_DIRECTORY#.

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.
- jQuery 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

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.

### 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**
- **About Using JavaScript Code in a Plug-In**

#### 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:

   ```html
   ...
   #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.
Understanding jQuery and jQuery UI Support

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" on page 19-14.
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.
Tip:

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);

19-14 Oracle Application Express Application Builder User's Guide


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.

```javascript
(function($) {
    
})(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.

**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:

```javascript
apex.event.trigger( "#myRegionStaticID", "apexrefresh" );
```
Managing Application Security

This section describes how to implement best practices for Oracle Application Express application security. While administrators are primarily responsible for ensuring the security of the Oracle Application Express installation, developers are responsible for building secure applications.

- Understanding Administrator Security Best Practices
- Understanding Developer Security Best Practices
- Controlling Access to Applications, Pages, and Page Components
- Establishing User Identity Through Authentication
- Providing Security Through Authorization

See Also: "Application Builder Concepts" on page 2-1 and "Understanding Application Builder" on page 4-1

Understanding Administrator Security Best Practices

Tip: This section references many settings in the 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
- About Configuring Oracle REST Data Services with Oracle Application Express
- About Configuring Oracle HTTP Server with mod_plsql with Oracle Application Express
- About Security Considerations When Using the Embedded PL/SQL Gateway
- Managing Instance Security
- Configuring Instance Settings
- About Configuring Workspace Purge Settings
- Understanding Workspace Management
- About Integrating with Oracle BI Publisher
- About the Advantages of the Application Express Runtime Environment
Enabling Network Services in Oracle Database 11g or Later

See Also: "Understanding Developer Security Best Practices" on page 20-14

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: Oracle Application Express Administration Guide for more information workspace and instance administration

About Configuring Oracle REST Data Services with Oracle Application Express

Oracle REST Data Services (formerly known as 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. To learn more, see "Restricting Access to Oracle Application Express by Database Access Descriptor (DAD)" in Oracle Application Express Administration Guide.

About Configuring Oracle HTTP Server with mod_plsql with Oracle Application Express

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 www_flow_epg_include_modules.authorize.

The purpose of the PlsqlRequestValidationFunction parameter is to control which procedures can be invoked through mod_plsql. By default, the only procedures permitted are the public entry points of Oracle Application Express. This can be extended using the validation functions shipped with Oracle Application Express. To learn more, see “Restricting Access to Oracle Application Express by Database Access Descriptor (DAD)” in Oracle Application Express Administration Guide.

About Security Considerations When Using the Embedded PL/SQL Gateway

The embedded PL/SQL gateway runs in the database as part of the Oracle XML DB HTTP listener. The Oracle XML DB HTTP listener and embedded PL/SQL gateway provides the equivalent core features of Oracle HTTP Server and mod_plsql. Because the HTTP Listener runs in the same database where Oracle Application Express is installed, it is not possible to separate the HTTP listener from the database. For this reason, Oracle does not recommend the embedded PL/SQL gateway for applications that run on the Internet or for production applications. Oracle recommends using Oracle REST Data Services instead. Additionally, the embedded PL/SQL gateway does not provide the same flexibility of configuration and detailed logging as Oracle REST Data Services.

Managing Instance Security

This section provides a summary of security best practices when managing an Oracle Application Express instance.

- About Creating Login Controls
- About Enabling Public File Upload
- About Restricting User Access by IP Address
- About Specifying an Instance Proxy
- About Utilizing Secure Sockets Layer
- About Enabling RESTful Access
- About Rejoin Sessions
- About Isolating Workspaces
- About Utilizing Session Timeout
- Restricting Password Reuse
- About Using SQL in Websheets
- About Enabling RESTful Services
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.

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

About Restricting User Access by IP Address
Administrators can restrict user access to an Oracle Application Express instance by entering a comma-delimited list of allowable IP addresses in the Restrict Access by IP Address attribute on the Manage Instance, Security page.

See Also: "Restricting User Access by IP Address" in Oracle Application Express Administration Guide

About Specifying an Instance Proxy
Administrators can use the Instance Proxy attribute to configure an entire Oracle Application Express instance to use a proxy for all outbound HTTP traffic. Setting a proxy at the instance-level supersedes any proxies defined at the application-level or in web service references. If a proxy is specified, regions of type URL, Web services, and report printing will use the proxy.

See Also: "Configuring a Proxy Server for an Instance" in Oracle Application Express Administration Guide
About Utilizing Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol for managing the security of data transmitted on the Internet. For web applications, SSL is implemented by using the HTTPS protocol. Oracle recommends that you run Oracle Application Express applications using SSL (HTTPS protocol) to prevent any sensitive data from being sent over an unencrypted (cleartext) communication channel.

Instance administrators can configure both their Oracle Application Express instance and all related applications to require HTTPS by configuring the **Require HTTPS** and **Require Outbound HTTPS** attributes. Configuring these attributes forces authenticated pages within the Application 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, Application 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*

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*

About Rejoin Sessions

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.

**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" on page 20-6.

## 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 countermeasure 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" and "Isolating All Workspaces in an Instance" in Oracle Application Express Administration Guide

## About Utilizing Session Timeout

By configuring Session Timeout attributes, you can reduce your application’s exposure. Users often leave their computers unattended for extended periods and do not close applications before departing. Therefore, an unauthorized person can easily assume the user’s identity within the application. By setting the session and idle timeout, users are automatically logged out of their application after the specified timeout.

Session Timeout attributes include:

- Maximum Session Length in Seconds
- Session Timeout URL
- Maximum Session Idle Time in Seconds
- Idle Timeout URL

Developers can configure session timeout for a specific application, or administrators can configure sessions timeout for an entire instance.

**See Also:** "Session Management" on page 7-21 to learn how to configure these attributes at an application-level. To configure these attributes for an entire Oracle Application Express instance, see "Configuring Session Timeout" in Oracle Application Express Administration Guide.

## Restricting Password Reuse

Site administrators can restrict password reuse for all administrator, developer, and end-user accounts based on a history of passwords previously used for the accounts.
To restrict password reuse:

1. Start SQL*Plus and connect to the database where Oracle Application Express is installed having the apex_administrator_role role or as SYS specifying the SYSDBA role. For example:
   - On Windows:
     ```sql
     SYSTEM_DRIVE:\ sqlplus /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```
   - On UNIX and Linux:
     ```sql
     $ 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:
   ```sql
   ALTER SESSION SET CURRENT_SCHEMA = APEX_050000;
   ```

3. Execute the following statement:
   ```sql
   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:
   ```sql
   BEGIN
   APEX_INSTANCE_ADMIN.SET_PARAMETER('PASSWORD_HISTORY_DAYS',0);
   COMMIT;
   END;
   /
   ```
   
   **Tip:** Developers can also use the APEX_UTIL.STRONG_PASSWORD_CHECK procedure and the APEX_UTIL.STRONG_PASSWORD_VALIDATION function described in Oracle Application Express API Reference.

**About Using SQL in Websheets**

Disabling the Allow SQL Websheet attribute enables administrators to prevent Websheet users from accessing underlying database objects using SQL tags or from creating SQL reports.

**See Also:** "Enabling and Disabling SQL Access in Websheets" in Oracle Application Express Administration Guide
About Enabling RESTful Services
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

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
■ About Enabling and Configuring Email
■ About Configuring Storage Options
■ About Creating Wallets

See Also: "Configuring Instance Settings" in Oracle Application Express Administration Guide

About Enabling 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

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 11gR2 or later.

See Also: "Configuring Email" in Oracle Application Express Administration Guide and "When and Why Network Services Must be Enabled" on page 20-11

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 are a specified number of days.
About Creating Wallets
A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL. You must create a wallet if you:

- Call a SSL-enabled URL (for example, by invoking a Web service).
- Create a region of type URL that is SSL-enabled.
- Configure secure SMTP by setting the Use SSL/TLS attribute to Yes.
- Have applications with LDAP authentication schemes that are configured to use SSL with Authentication.

See Also: "Configuring Wallet Information" in Oracle Application Express Administration Guide

About Configuring Workspace Purge Settings
Inactive workspaces consume valuable storage space and degrade system performance. By enabling Workspace Purge Settings, administrators can configure Oracle Application Express to purge inactive workspaces.

See Also: "Purging Inactive Workspaces" in Oracle Application Express Administration Guide

Understanding Workspace Management
This section provides a summary of security best practices when managing workspaces in Oracle Application Express.

- About Managing Workspace to Schema Assignments
- About Managing Developers and Users
- About Managing Component Availability

See Also: "Managing Existing Workspaces" in Oracle Application Express Administration Guide

About Managing Workspace to Schema Assignments
When users log in to Oracle Application Express, they log in to a shared work area called a workspace. Each workspace can have multiple associated (or mapped) schemas. Instance administrators can define associations between workspaces and schemas as appropriate. By associating a workspace with a schema, developers in that workspace can create new database objects in that schema and build applications that interact with the database objects in that schema.

See Also: "Managing Workspace to Schema Assignments" in Oracle Application Express Administration Guide

About Managing Developers and Users
Administrators can define what schemas specific users have access to and also their user role, that is if they are a workspace administrator, a developer, or an end user. Administrators can also limit developer access to specific Application Express
components and lock accounts. Oracle Application Express includes two separate interfaces for managing developers and users. Workspace administrators manage user accounts within their workspace and Instance administrators manage all user accounts across an Application Express development instance.

**See Also:** "Managing Users Across an Application Express Instance" and "Managing Users in a Workspace" in *Oracle Application Express Administration Guide*

### About Managing Component Availability

Workspace administrators can control user access to key components such as Application Builder, Team Development, SQL Workshop, and PL/SQL editing, which determines whether developers have the ability to edit and compile PL/SQL program units from Object Browser. For example, suppose an administrator wants users to be able to build database components, run SQL statements, but not create applications, he or she could define a workspace with rights to a specific schema and then configure the users as developers using the options on the Set Workspace Preferences page.

**See Also:** "Configuring Workspace Preferences" in *Oracle Application Express Administration Guide*

### About Integrating with Oracle BI Publisher

Oracle recommends that you employ Secure Sockets Layer (SSL) when integrating Oracle Application Express with Oracle BI Publisher. Once SSL (HTTPS protocol) is configured for your Oracle BI Publisher server, you must create a wallet and specify the HTTPS protocol for the Report Server in the internal administration Environment Settings.

**See Also:** "Configuring Wallet Information" and "Configuring Report Printing" in *Oracle Application Express Administration Guide*

### About the Advantages of the Application Express Runtime Environment

An Oracle Application Express runtime environment enables you to run production applications, but does not provide a web interface for administration or direct development of these applications.

Oracle recommends that you run any sensitive production Oracle Application Express applications with a runtime installation of Oracle Application Express. A runtime installation does not expose the web-based application development environment, thus preventing the use of Application Builder, SQL Workshop, and related utilities on a production installation. Additionally, a runtime environment only includes the Oracle Application Express database objects and privileges necessary to run applications, making it a more hardened environment.

**See Also:** "Installing Exported Applications into a Runtime Environment" in *Oracle Application Express Administration Guide*

### Enabling Network Services in Oracle Database 11g or Later

This section describes how to enable network services in Oracle Database 11g or later.

- **When and Why Network Services Must be Enabled**
- **Granting Connect Privileges Prior to Oracle Database 12c**
- **Granting Connect Privileges in Oracle Database 12c**
■ About Troubleshooting an Invalid ACL Error

When and Why Network Services Must be Enabled
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_050000` 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 Oracle Database 11g or 12c will already have the parameter set properly, but a database upgraded to Oracle Database 12c from a prior version may not. See “Creating and Configuring an Oracle Database” in Oracle Database Administrator’s Guide for information about changing database initialization parameters.

Granting Connect Privileges Prior to Oracle Database 12c
The following example demonstrates how to grant connect privileges to any host for the `APEX_050000` database user. This example assumes you connected to the database where Oracle Application Express is installed as `SYS` specifying the `SYSDBA` role.

```sql
DECLARE
  ACL_PATH  VARCHAR2(4000);
  ACL_ID    RAW(16);
BEGIN
  -- Look for the ACL currently assigned to '*' and give APEX_050000
  -- the 'connect' privilege if APEX_050000 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;

  -- Before checking the privilege, ensure that the ACL is valid
  -- (for example, does not contain stale references to dropped users).
  -- If it does, the following exception will be raised:
  --
  -- ORA-44416: Invalid ACL: Unresolved principal 'APEX_050000'
  -- ORA-06512: at "XDB.DBMS_XDBZ", line ...
  --
  SELECT SYS_OP_R2O(extractValue(P.RES,'/Resource/XMLRef')) INTO ACL_ID
  FROM XDB.XDB$ACL A, PATH_VIEW P
  WHERE extractValue(P.RES,'/Resource/XMLRef') = REF(A) AND
          EQUALS_PATH(P.RES, ACL_PATH) = 1;
  DBMS_XDBZ.ValidateACL(ACL_ID);
  IF DBMS_NETWORK_ACL_ADMIN.CHECK_PRIVILEGE(ACL_PATH, 'APEX_050000','connect') IS NULL THEN
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(ACL_PATH, 'APEX_050000', TRUE, 'connect');
  END IF;
END;
```

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 Oracle Database 11g or 12c will already have the parameter set properly, but a database upgraded to Oracle Database 12c from a prior version may not. See “Creating and Configuring an Oracle Database” in Oracle Database Administrator’s Guide for information about changing database initialization parameters.
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);
  ACL_ID    RAW(16);
BEGIN
  -- Look for the ACL currently assigned to 'localhost' and give APEX_050000
  -- the 'connect' privilege if APEX_050000 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;

  -- Before checking the privilege, ensure that the ACL is valid
  -- (for example, does not contain stale references to dropped users).
  -- If it does, the following exception will be raised:
  --
  -- ORA-44416: Invalid ACL: Unresolved principal 'APEX_050000'
  -- ORA-06512: at 'XDB.DBMS_XDBZ', line ...
  --
  SELECT SYS_OP_R2O(extractValue(P.RES, '/Resource/XMLRef')) INTO ACL_ID
  FROM XDB.XDB$ACL A, PATH_VIEW P
  WHERE extractValue(P.RES, '/Resource/XMLRef') = REF(A) AND
    EQUALS_PATH(P.RES, ACL_PATH) = 1;

  DBMS_XDBZ.ValidateACL(ACL_ID);
  IF DBMS_NETWORK_ACL_ADMIN.CHECK_PRIVILEGE(ACL_PATH, 'APEX_050000', 'connect') IS NULL THEN
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(ACL_PATH, 'APEX_050000', 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 power users to connect to everywhere',
      'APEX_050000', TRUE, 'connect');
    DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL('local-access-users.xml','localhost');
END;
/ 
COMMIT;
```
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 using APPEND_HOST_ACE instead. The following example demonstrates how to grant connect privileges to any host for the APEX_050000 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_050000',
                            principal_type => xs_acl.ptype_db));
END;
/

The following example demonstrates how to provide less privileged access to local network resources. This example enables access to servers on the local host only, such as email and report servers.

BEGIN
    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
        host => 'localhost',
        ace => xs$ace_type(privilege_list => xs$name_list('connect'),
                          principal_name => 'apex_050000',
                          principal_type => xs_acl.ptype_db));
END;
/

About Troubleshooting an Invalid ACL Error

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);

    -- Remove invalid users or drop the ACL.
    IF CNT > 0 THEN
        UPDATE XDS_ACE
        SET ACE.ACLID = NULL
        WHERE ACLID = ACL_ID;
    ELSE
        DROP_ACL(ACL_ID);  
    END IF;
END;
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 =
                DELETEMAIL(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_050000 user. See "Granting Connect Privileges Prior to Oracle
Database 12c" on page 20-11.

Understanding Developer Security Best Practices

This section describes security best practices for Oracle Application Express
developers.

■ About Items of Type Password
■ Identifying At Risk Password Items
■ Understanding Cross-Site Scripting Protection
■ About Session State and Security
■ Understanding Session State Protection
■ About Securing File Uploads

See Also: "Understanding Administrator Security Best Practices" on
page 20-1 and "About Application Express User Roles" on page 1-3

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” on page 20-20.

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

■ Viewing the Password Items Report

Viewing the Security Profiles Report

To view the Security Profiles Report:

1. Navigate to the Workspace home page.

2. Click the Application Builder icon.

   The Application 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:
Understanding Developer Security Best Practices

- Application
- Name
- Parsing Schema
- Application Level Authorization Scheme
- Authentication
- Authorization Schemes
- Authorization Schemes
- Pages
- Encrypted Items
- At Risk Password Items

Viewing the Password Items Report
To identify at risk password items:

1. Navigate to the Workspace home page.
2. Click the Application Builder icon.
   The Application Builder home page appears.
3. Click the Workspace Utilities icon.
4. Locate Cross Application Reports on the right side of the window.
5. Under Cross Application Reports, click Password Items.
   The Password Items report shows all of the password items within the application and indicates if they use encryption and whether they save state. Password items that do neither are highlighted as At Risk.

   Tip: For pages that contain password items, set the page attribute Form Auto Complete to Off. Setting that attribute to Off prevents the web browser from attempting to auto complete items on the page.

Understanding Cross-Site Scripting Protection
This section explains best practices for protecting your application from a cross site-scripting security breach.

- What Is Cross Site-scripting Security Breach?
- Protecting HTML Regions and Other Static Areas
- About Protecting Dynamic Output
- About Protecting Report Regions
- About Protecting Form Items
- About Restricting Characters Entered on an Item

What Is Cross Site-scripting Security Breach?
Cross site-scripting (also referred to as XSS) is a security breach that takes advantage of dynamically generated web pages. In a XSS attack, a web application is sent a script that activates when it is read by a user's browser. Once activated, these scripts can steal data, even session credentials, and return the information to the attacker.
If malicious code were introduced into an Oracle Application Express application, it could be rendered into HTML regions and other places within the application during normal page rendering. To prevent the introduction of malicious code into session state, the Application Express engine escapes characters in certain cases.

**Protecting HTML Regions and Other Static Areas**
This section explains how to protect HTML regions and other static display areas.

- **About Protecting HTML Regions and Other Static Areas**
- **About Safe Item Display Types**
- **About the Rules Used to Determine Whether to Escape Values**
- **About Using Safe Item Types to Hold and Emit HTML Markup**

**About Protecting HTML Regions and Other Static Areas** In HTML regions and other static display areas, you can reference session state using the substitution strings. Special substitution strings available within a template are denoted by the number symbol (#), for example, #ABC#. To reference page or application items use the &ITEM. notation.

Examples of static display areas include HTML regions, page headers and footers, region headers and footers, region titles, button labels, help text, form item labels and post-element text, templates, radiogroup (before and after field text), event success messages, event error messages, navigation bar attributes, application static substitution string values, chart labels and legends, breadcrumbs and list framing text, and calendar text, labels, or legends.

Developers can also append an exclamation mark (!) followed by a predefined filter name to a page or application item name or to a report column reference to escape special characters in the substitution value.

**See Also:** "About Using Substitution Strings" on page 2-25 and "Controlling Output Escaping in Substitution Strings" on page 2-26

**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" on page 15-1

**About the Rules Used to Determine Whether to Escape Values** The Application Express engine uses predefined smart escaping rules to determine if and when to escape values fetched from session state.

The reason for these rules is that items that use the display types listed previously are often for text containing HTML that is intended to be emitted to the browser without being filtered (that is, escaped). The only way this can be made safe is by the enforcement of the rule that these types of items are always escaped on input to the application. For example, if a user passes some text into a safe item using an Oracle Application Express f?p URL syntax, the Application Express engine escapes special characters when saving the value into session state. This has two intended results:
1. If the value contained no special characters, the value passed in is saved into session state exactly as it was provided.

2. If the value contained special characters, those characters are escaped when the value is saved into session state.

In either situation, the item can now safely be referenced using an &ITEM. notation in any HTML region or other static area mentioned previously.

**About Using Safe Item Types to Hold and Emit HTML Markup** You can use the safe item types listed previously to hold and emit HTML markup to the browser. For example, suppose you have a requirement to render some text in bold face by referencing a safe page item named P1_XXX (using &P1_XXX.) The item P1_XXX is presumed to contain the following HTML:

\(<b>ABABABAB</b>\)

You can achieve this by using application controls (computations, processes, item source expressions, item default values, and so on) to store values into these safe items. When values are introduced in this way, you ensure the safety of the content. When you use these methods, the Application Express engine does not escape any special characters when saving the values into session state.

Finally, the safety of safe items is ensured by a rule that prevents those items from being posted on a page and submitted to the Application Express engine as part of a page submission.

**About Protecting Dynamic Output**

Items fetched from session state and rendered using `htp.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:

```
htp.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:

```
htp.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 described in "About Safe Item Display Types" on page 20-17.

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 an XSS attack using your application if you do not follow this rule.

**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:

```sql
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:

```html
<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# instead, 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" on page 2-26, "Editing Column Attributes in Page Designer" on page 10-12, and "Editing Column Attributes in Component View" on page 10-35

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" on page 2-26
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.


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 Application 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.
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**
- **Enabling Session State Protection**
- **Configuring Session State Protection**

**How Session State Protection Works**
When enabled, Session State Protection uses the Page Access Protection attributes and the Session State Protection item attributes with checksums positioned in `f?p=` URLs to prevent URL tampering and unauthorized access to and alteration of session state. When Session State Protection is disabled, the page and item attributes related to session state protection are ignored and checksums are not included checksums in generated `f?p=` URLs.

**Enabling Session State Protection**
You can enable session state protection from either the Edit Security Attributes page or the Session State Protection page.

Enabling Session State Protection is a two-step process. First, you enable the feature. Second, you set page and item security attributes. You can perform these steps using a wizard, or you can set security attributes for pages and items manually on the Session State Protection page.

- **Enabling Session State Protection from Edit Security Attributes**
- **About the Expire Bookmarks Button**
- **Enabling Session State Protection from Session State Protection**

**Enabling Session State Protection from Edit Security Attributes**

To enable Session State Protection from the Edit Security Attributes page:

1. On the Workspace home page, click the **Application 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** 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 checkums or contain unnecessary checksums are not affected by Session State Protection.

During page rendering, the Application Express engine uses a hidden application attribute (a checksum salt) during computation and to verify checksums included in f?p URLs. When you enable Session State Protection, the Application Express engine includes checksums. You can reset the checksum salt attribute by clicking **Expire Bookmarks** on the Edit Security Attributes page.

   **Tip:** Be aware that if you click **Expire Bookmarks**, bookmarked URLs used to access your application that contain previously generated checksums will fail.

On the Confirm Invalidate Bookmarks page, you can also configure the Bookmark Checksum Function which is the hash algorithm used for computing bookmark checksums.

**Enabling Session State Protection from Session State Protection** To enable Session State Protection:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click the **Application Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Security, select **Session State Protection**.

   The Session State Protection page appears. Note the current Session State Protection status (Enabled or Disabled) displays at the top of the page.

2. Click the **Set Protection** button.

   The Session State Protection wizard appears.

3. Under Select Action, select **Enable** and click **Next**.
Next, determine whether to set security attributes for pages and items.

4. Select **Enable** and click **Next**.

5. Click **Enable Session State Protection**.

   **Tip:** To disable Session State Protection, perform the same steps, but select **Disable** instead of **Enable**. Disabling Session State Protection does not change existing security attribute settings, but those attributes are ignored at runtime.

### Configuring Session State Protection

- **About Configuring Session State Protection**
- **Reviewing Existing Session State Protection Settings**
- **Configuring Session State Protection Using a Wizard**
- **Configuring Session State Protection for Pages**
- **Configuring Session State Protection for Page Items**
- **Configuring Session State Protection for Application Items**

   **Tip:** Before you can configure security attributes, you must first enable Session State Protection. See "Enabling Session State Protection" on page 20-21.

### About Configuring Session State Protection

Once you have enabled Session State Protection, the next step is to configure security attributes. You can configure security attributes in two ways:

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

### Reviewing Existing Session State Protection Settings

You can review a summary of Session State Protection settings for pages, items, and application items on the first page of the Session State Protection wizard.

To view summaries of existing Session State Protection settings:

1. Navigate to the Session State Protection page:
   a. On the Workspace home page, click the **Application Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Security, select **Session State Protection**.

      The Session State Protection page appears.

2. Click **Set Protection**.

3. Expand the following reports at the bottom of the page:
   - Page Level Session State Protection Summary
   - Page Item Session State Protection Summary
   - Application Item Session State Protection
Configuring Session State Protection Using a Wizard When you configure Session State Protection using a wizard, you set a value for specific attribute categories. Those selections are then applied to all pages and items within the application.

To configure Session State Protection using a wizard:

1. Navigate to the Session State Protection page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
      The Session State Protection page appears.

2. Click Set Protection.
   The Session State Protection wizard appears.

3. Under Select Action, select Configure and click Next.

4. For Page Access Protection, select one of the following:
   ■ Unrestricted - The page may be requested using a URL with or without session state arguments (Request, Clear Cache, Name/Values).
   ■ Arguments Must Have Checksum - If Request, Clear Cache, or Name/Value arguments appear in the URL, a checksum must also be provided. The checksum type must be compatible with the most stringent Session State Protection attribute of all the items passed as arguments.
   ■ No Arguments Allowed - A URL may be used to request the page but no Request, Clear Cache, or Name/Value arguments are allowed.
   ■ No URL Access - The page may not be accessed using a URL; however, the page may be the target of a Branch to Page branch type, which does not do a URL redirect.

5. For Application Item Protection, select one of the following:
   ■ Unrestricted - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
      Note: If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).
   ■ Checksum Required: Application Level - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.
   ■ Checksum Required: User Level - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same
named user, running the same application in the current workspace but in a different session.

- **Checksum Required: Session Level** - The item’s session state may be set by passing the item name/value in a URL if a checksum specific to the current session is also provided. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.

- **Restricted - May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this option when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is applicable only to items that cannot be used as data entry items and is always observed even if Session State Protection is disabled.

Use this attribute for application items or for page items with any of these Display As types:
- Display as Text (escape special characters, does not save state)
- Display as Text (does not save state)
- Display as Text (based on LOV, does not save state)
- Display as Text (based on PLSQL, does not save state)
- Text Field (Disabled, does not save state)
- Stop and Start HTML Table (Displays label only)

6. For Page Data Entry Item Protection, select one of the following:

- **Unrestricted** - The item’s session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.

  **Note:** If you must set this item’s value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).

- **Checksum Required: Application Level** - The item’s session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.

- **Checksum Required: User Level** - The item’s session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum (see next bullet). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same named user, running the same application in the current workspace but in a different session.

- **Checksum Required: Session Level** - The item’s session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.

7. For Page Display-Only Item Protection, select one of the following:
- **Unrestricted** - The item may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.

  *Note:* If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).

- **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. You can also use a user-level checksum or a session-level checksum (see next bullets). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.

- **Checksum Required: Session Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.

- **Checksum Required: User Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. You can also use a session-level checksum. Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same named user, running the same application in the current workspace but in a different session.

- **Restricted: May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is always observed, even if Session State Protection is disabled.

  This attribute may be used with any of these Display As types:
  - Display as Text (escape special characters, does not save state)
  - Display as Text (does not save state)
  - Display as Text (based on LOV, does not save state)
  - Display as Text (based on PLSQL, does not save state)
  - Text Field (Disabled, does not save state)
  - Stop and Start HTML Table (Displays label only)

8. Click **Next**.

9. Click **Finish**.

**Configuring Session State Protection for Pages** To configure Session State Protection for Pages:

1. Navigate to the Session State Protection page:
   a. On the Workspace home page, click the **Application 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. To learn more, see "About the Search Bar" on page 10-57.

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**.

**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 **Application 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. To learn more, see "About the Search Bar" on page 10-57.

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.

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 Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
      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. To learn more, see “About the Search Bar” on page 10-57.

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.
Controlling Access to Applications, Pages, and Page Components

- **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**.

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" on page 10-89 or use the methods described in Oracle Application Express How To Documents for file upload on OTN at:


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" on page 15-1 and "About Item Types" on page A-1 to learn more about creating a File Browse page-level item.

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**
- **Creating an Access Control List**
- **Selecting an Application Mode and Adding Users**
- **About Controlling Access for Pages and Page Components**

**See Also:** "Building Queries with Query Builder" in Oracle Application Express SQL Workshop Guide

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:
Controlling Access to Applications, Pages, and Page Components

- **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" on page 20-52

**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
- Creating an Access Control While Editing a Page

**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 Application 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. 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.

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:
   a. Page Designer - Click the Create menu at the top of the page and select Page.
   b. 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.
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.

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
- Adding Users to an Access Control List and Selecting Privileges
- Removing Users from the Access Control List

Selecting an Application Mode
To select an application mode:

   The wizard creates a page named Access Control Administration.
2. Run the Access Control Administration page. See "Running an Application or Page" on page 2-9.
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](image)

   The user interface of your page depends on the theme you selected for your application. See "Using Themes" on page 13-1.

   Next, add users to the Access Control List.

   **Adding Users to an Access Control List and Selecting Privileges**

   To add users to the Access Control List:

      The wizard creates a page named Access Control Administration.

   2. Run the Access Control Administration page. See "Running an Application or Page" on page 2-9.

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.

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.

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.
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
- Creating an Authentication Scheme
- Understanding Preconfigured Authentication Schemes
- Managing Existing Authentication Schemes
- Viewing the Authentication Scheme Associated with an Application
- Changing the Authentication Scheme Associated with an Application
- Creating a Login Page
- Viewing Authentication Scheme Reports

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
- About Support for Deep Linking
- About Determining Whether to Include Authentication

How Authentication Works

You determine how your application interacts with users. If all users have the same rights and privileges, they are referred to as public users. However, if your application must track each user individually, you must specify an authentication method.

Authentication establishes the identity of each user who accesses your application. Many authentication processes require that a user provide some type of credentials such as a user name and password. These credentials are then evaluated and they either pass or fail. If the credentials pass, the user has access to the application. Otherwise, access is denied.

Once a user has been identified, the Application Express engine keeps track of each user by setting the value of the built-in substitution string \texttt{APP\_USER}. As a user navigates from page to page, the Application Express engine sets the value of \texttt{APP\_USER} to identify the user. The Application Express engine uses \texttt{APP\_USER} as one component of a key for tracking each user's session state.

From a programming perspective, you can access \texttt{APP\_USER} using the following syntax:

\texttt{See Also: "Attaching an Authorization Scheme to an Application, Page, or Components" on page 20-52}
Establishing User Identity Through Authentication

- 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:

```sql
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" on page 2-34 and "Configuring Security Attributes" on page 7-19

About Support for Deep Linking

Oracle Application Express applications that use authentication schemes support deep linking. Deep linking refers to the ability to link to an Oracle Application Express page out of context (for example, from a hyperlink in an email or workflow notification). When you link to a page out of context and the application requires the user be authenticated, the user is taken to the login page. After credentials verification, the Application Express engine automatically displays the page that was referenced in the original link.

About Determining Whether to Include Authentication

As you create your application, you must determine whether to include authentication. You can:

- **Choose to not require authentication.** Oracle Application Express does not check any user credentials. All pages of your application are accessible to all users.

- **Select a built-in authentication scheme.** Create an authentication method based on available preconfigured authentication schemes. Depending on which scheme you choose, you may also have to configure the corresponding components of Oracle 10g iAS, Oracle Internet Directory, or other external services. See "Understanding Preconfigured Authentication Schemes" on page 20-39 and "Changing the Authentication Scheme Associated with an Application" on page 20-48.

- **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 "Creating and Editing an Authorization Scheme" on page 20-49.
Creating an Authentication Scheme

To create an authentication scheme:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
   The Authentication Schemes page appears.
5. 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" on page 10-57.
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. See "Understanding Preconfigured Authentication Schemes" on page 20-39 and "Custom Authentication" on page 20-40.
   - As a copy of an existing authentication scheme.
8. Follow the on-screen instructions.

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.

- Application Express Accounts
- Custom Authentication
- Database Accounts
- HTTP Header Variable
- LDAP Directory Verification
- No Authentication (using DAD)
- Open Door Credentials
- Oracle Application Server Single Sign-On Server

**Tip:** Authentication schemes support plug-ins. To learn more, see "Implementing Plug-ins" on page 19-3.

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
- Setting Up Application Express Accounts Authentication
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

Setting Up Application Express Accounts Authentication  To set up Application Express Accounts:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
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.

Custom Authentication
This section explains when you might want to use the Custom Authentication and explains how to configure it.
- About Custom Authentication
- Setting Up Custom Authentication
- About Session Management Security

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" on page 19-3.

**Setting Up Custom Authentication** To create a custom authentication scheme:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
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.
9. 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. See "Viewing Field-Level Help" on page 1-21.
10. Click Create.

**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.
Establishing User Identity Through Authentication

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
- Setting Up Database Account Credentials

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.

Setting Up Database Account Credentials To set up Database Account Credentials:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
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.

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
- Setting Up HTTP Header Variable

About HTTP Header Variable HTTP Header Variable supports the use of header variables to identify a user and to create an Application Express user session. Use 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).

Setting Up HTTP Header Variable To set up HTTP Header Variable:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
The Shared Components page appears.


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.

9. To learn more about a specific field, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

10. Click Create.

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
■ Setting Up LDAP Directory Verification

About LDAP Directory Verification You can configure any authentication scheme that uses a login page to use Lightweight Directory Access Protocol (LDAP) to verify the user name and password submitted on the login page.

Application 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 call for credentials verification. When you create an LDAP Credentials authentication scheme, the wizard requests and saves the LDAP host name, LDAP port, DN string, and determines whether to use SSL, exact DN, and optionally a search filter if not using exact DN. An optional preprocessing function can be specified to adjust formatting of the user name passed to the API.

Setting Up LDAP Directory Verification To set up LDAP Directory:

1. On the Workspace home page, click the Application Builder icon.

2. Select an application.

3. On the Application home page, click Shared Components.

   The Shared Components page appears.


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.

9. To learn more about a specific field, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

10. Click Create.

---

**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 "Configuring Wallet Information" in *Oracle Application Express Administration Guide* and "Using Oracle Wallet Manager" in *Oracle Database Advanced Security Guide*.

---

**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**
- **Setting Up DAD Credentials Verification**
About DAD Credentials Verification

DAD credentials verification uses the Oracle database native authentication and user mechanisms to authenticate users using a basic authentication scheme. This authentication scheme gets the user name from the DAD either as the value stored in the DAD configuration or, if the account information is not stored in the DAD configuration, as the user name captured using the basic authentication challenge.

To use DAD credentials verification:

- Each application user must have a user account in the Oracle database.
- You must configure a PL/SQL DAD for basic authentication (without account information).

This results in one user name/password challenge for browser session for your application users. The user identity token is then made available in the APP_USER item.

DAD database authentication is useful when you must implement an authentication method that requires minimal setup for a manageable number of users. Ideally these users would have self-managed accounts in the database and your use of this authentication method would be short lived (for example, during the demonstration or prototyping stages of development).

The main drawback of this approach is burdensome account maintenance, especially if users do not administer their own passwords, or if their database accounts exist only to facilitate authentication to your application.

Setting Up DAD Credentials Verification

To set up DAD Credentials Verification:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
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 (using DAD).
8. Click Create.

Open Door Credentials

This section explains when you might want to use Open Door Credentials and explains how to configure it.

- About Door Credentials
- Setting Up Open Door Credentials

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.
**Setting Up Open Door Credentials** To set up Open Door Credentials:

1. On the Workspace home page, click the **Application 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**.

**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**
- **Setting Up Oracle Application Server Single Sign-On**

**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.

**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 **Application 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.

9. To learn more about a specific field, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

10. Click **Create**.

**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 **Application 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. To learn more, see "About the Search Bar" on page 10-57.

6. To edit a specific authentication scheme as a report, click the Edit icon the to the left of the authentication scheme name.

   The Authentication Scheme page appears.

7. To learn more about a specific field, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

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

**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 **Application 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**.
Establishing User Identity Through Authentication

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 Application Builder icon.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Shared Components page appears.

2. Click the Edit icon adjacent to the authentication scheme you want to activate.
   The Create/Edit page appears.

3. Click Make Current Scheme.

   See Also: "Understanding Preconfigured Authentication Schemes" on page 20-39

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 Application Builder icon.
2. Select an application.
3. Click Create Page.
4. For Create a Page:
   a. User Interface - Select a user interface for the page.
   b. Select a page type - Select Login Page.
   c. Click Next.

5. Select Login Page.
6. Specify Login page attributes and click Create.

Viewing Authentication Scheme Reports

To view authentication scheme reports:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
5. Click the appropriate tab at the top of the page:
   ■ **Subscription** - Use the Authentication Scheme Subscription report to view subscribed authentication schemes in your application.
   ■ **History** - Use the Authentication Scheme History report to view recent modifications made to items in this application.

### Providing Security Through Authorization

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**
- **Creating and Editing an Authorization Scheme**
- **Attaching an Authorization Scheme to an Application, Page, or Components**
- **Viewing Authorization Reports**

#### 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.

#### 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.

- **Creating an Authorization Scheme**
- **Editing Attributes of an Existing Authorization Scheme**
- **Changing the Evaluation Point Attribute**
- **About Resetting Authorization Scheme State**
Providing Security Through Authorization

About Authorization Scheme Types
When you create an authorization scheme you select an authorization scheme type. The authorization scheme type determines how an authorization scheme is applied. Developers can create new authorization type plug-ins to extend this list.

Table 20–1 Authorization Scheme Types

<table>
<thead>
<tr>
<th>Authorization Scheme Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists SQL Query</td>
<td>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.</td>
</tr>
<tr>
<td>NOT Exists SQL Query</td>
<td>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.</td>
</tr>
<tr>
<td>PL/SQL Function Returning Boolean</td>
<td>Enter a function body. If the function returns true, the authorization succeeds.</td>
</tr>
<tr>
<td>Item in Expression 1 is NULL</td>
<td>Enter an item name. If the item is null, the authorization succeeds.</td>
</tr>
<tr>
<td>Item in Expression 1 is NOT NULL</td>
<td>Enter an item name. If the item is not null, the authorization succeeds.</td>
</tr>
<tr>
<td>Value of Item in Expression 1 Equals Expression 2</td>
<td>Enter an item name and value. The authorization succeeds if the item’s value equals the authorization value.</td>
</tr>
<tr>
<td>Value of Item in Expression 1 Does NOT Equal Expression 2</td>
<td>Enter an item name and a value. The authorization succeeds if the item’s value is not equal to the authorization value.</td>
</tr>
<tr>
<td>Value of Preference in Expression 1 Does NOT Equal Expression 2</td>
<td>Enter a preference name and a value. The authorization succeeds if the preference’s value is not equal to the authorization value.</td>
</tr>
<tr>
<td>Value of Preference in Expression 1 Equals Expression 2</td>
<td>Enter a preference name and a value. The authorization succeeds if the preference’s value equals the authorization value.</td>
</tr>
<tr>
<td>Is In Group</td>
<td>Enter a group name. The authorization succeeds if the group is enabled as a dynamic group for the session. See &quot;APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS&quot; in Oracle Application Express API Reference. 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.</td>
</tr>
<tr>
<td>Is Not In Group</td>
<td>Enter a group name. The authorization succeeds if the group is not enabled as a dynamic group for the session.</td>
</tr>
</tbody>
</table>

Creating an Authorization Scheme
To create an authorization scheme:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components. The Shared Components page appears.
5. Click Create.
6. Specify how to create an authorization scheme by selecting one of the following:
   - From Scratch
   - As a Copy of an Existing Authorization Scheme
7. On Create Authorization Scheme - Details:
   a. Name - Enter an unique name that identifies this authorization scheme.
   b. Scheme Type - Select how this authorization scheme will be applied. See "About Authorization Scheme Types" on page 20-50.
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 control if future uses cause re-evaluations and when a memorized result can be taken instead.

For more details, see field-level Help. See "Viewing Field-Level Help" on page 1-21.

8. Click Create Authorization Scheme.

See Also: "Attaching an Authorization Scheme to an Application, Page, or Components" on page 20-52, "Changing the Evaluation Point Attribute" on page 20-51, and "About Resetting Authorization Scheme State" on page 20-52

Editing Attributes of an Existing Authorization Scheme
To edit attributes of an existing authorization scheme:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
   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" on page 10-57.
5. Select an authorization scheme.
6. Edit the appropriate attributes.
   To learn more, see field-level Help. See "Viewing Field-Level Help" on page 1-21.
7. To save your changes, click Apply Changes.

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 Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   The Shared Components page appears.
   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" on page 10-57.
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. See " Viewing Field-Level Help" on page 1-21.

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

   **Tip:** The default value **Once per session** is the most efficient. You should choose another value if the authorization check depends on changing session state or other factors that are not consistent over an entire session.

### About Resetting Authorization Scheme State

If an authorization scheme is validated once for each session, Oracle Application Express caches the validation results in each user’s session cache. You can reset a session’s authorization scheme state by calling the `APEX_AUTHORIZATION.RESET_CACHE` API.

**See Also:** "APEX_AUTHORIZATION.RESET_CACHE Procedure" in Oracle Application Express API Reference

### Attaching an Authorization Scheme to an Application, Page, or Components

Once you have created an authorization scheme you can attach it to an entire application, page, control, or component.

- Attaching an Authorization Scheme to an Application
- Attaching an Authorization Scheme to a Page
- Attaching an Authorization Scheme to a Control or Component

#### Attaching an Authorization Scheme to an Application

To attach an authorization scheme to an application:

1. On the Workspace home page, click the **Application 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" on page 7-21.

7. To define a new authorization scheme, click Define Authorization Schemes.

**Attaching an Authorization Scheme to a Page**

To attach an authorization scheme to a page:

1. On the Workspace home page, click the Application 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.

**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 Application 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.

**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 Application Builder icon.
2. Select an application.
3. On the Application home page, click Shared Components.
   - The Shared Components page appears.
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.
Managing Application Globalization

You can run applications built in Application Builder concurrently in different languages. This section describes how to translate an application built in Application Builder.

- Understanding Application Translation and Globalization Support
- Specifying the Primary Language for an Application
- Understanding the Translation Process
- Translating Messages
- Translating Data That Supports List of Values
- Understanding Supported Globalization Codes

**See Also:** "Viewing Installed Translations" in Oracle Application Express Administration Guide

### Understanding Application Translation and Globalization Support

You can develop applications in Application 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
- About Language Identification
- About the Rules for Translating Database Applications
- How Translated Applications Are Rendered
- Making Application Attributes Translatable

### About the Translation Process

In general, translating an application built in Application Builder involves the following steps:

- Step 1 - Map the primary and target application IDs.
- Step 2 - Seed and export the text to a file for translation.
- Step 3 - Translate the text in the file.
- Step 4 - Apply the translated file.
- Step 5 - Publish the translated file.
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" on page 21-4

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.

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" on page 21-3 and "Translating Data That Supports List of Values" on page 21-34

Making Application Attributes Translatable

When you build an application in Application 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

- About Messages
About Dynamic Translation Text Strings

About Translating Region Titles

About Translating Templates

About Shortcuts that Support Translatable Messages

Application 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" on page 18-34

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" on page 21-13

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" on page 21-34

About Translating Region Titles

By default, page region titles are included in the generated translation file. However, you can mark a region title as not translatable.

- **Marking a Region as Not Translatable in Page Designer**

- **Marking a Region as Not Translatable in Page Designer**

  **Marking a Region as Not Translatable in Page Designer** To mark a region title as not translatable:


    Page Designer appears

  2. In the Rendering tree, 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.

Marking a Region as Not Translatable in Component View  To mark a region title as not translatable:
2. On the page, select the region title.
   The Edit Region page appears.
3. Under Identification, select the exclude title from translation check box.

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 Application 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" on page 13-23 and "Substitutions" on page 7-19

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 Application Builder icon.
2. Select an application.
3. Click Shared Components.
5. For Application Primary Language, select the language in which the application is being developed.
6. For **Application Language Derived From**, specify how the Application Express engine determines (or derives) the application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item. **Table 21–1** describes available options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No NLS (Application not translated)</td>
<td>Select this option if the application will not be translated.</td>
</tr>
<tr>
<td>Application Primary Language</td>
<td>Determines the application’s primary language based on the Application Primary Language attribute. (See step 5.)</td>
</tr>
<tr>
<td>Browser (use browser language preference)</td>
<td>Determines the application’s primary language based on the user’s browser language preference.</td>
</tr>
<tr>
<td>Application Preference (use FSP_ LANGUAGE_PREFERENCE)</td>
<td>Determines the application’s primary language based on a value defined using the APEX_UTIL.SET_PREFERENCE API. Select this option to maintain the selected language preference across multiple logins.</td>
</tr>
<tr>
<td>Item Preference (use item containing preference)</td>
<td>Determines the application’s primary language based on an application-level item called FSP_LANGUAGE_PREFERENCE. Using this option requires Oracle Application Express to determine the appropriate language preference every time the user logs in.</td>
</tr>
<tr>
<td>Session</td>
<td>Determines the translated application language from the session setting. The Application Express session language can be set through either the APEX_UTIL.SET_SESSION_LANG procedure or by using the p_lang parameter of the &quot;f&quot; procedure in the URL.</td>
</tr>
</tbody>
</table>

7. Configure other options as appropriate. To learn more about an attribute, see field-level Help.

8. Click **Apply Changes**.

   **See Also:** "Editing the Application Attributes" on page 7-14, "Configuring Security Attributes" on page 7-19, and "Understanding Supported Globalization Codes" on page 21-34

**Using Format Masks for Items**

This section describes how to use format masks for items.

- **About Apply Format Masks to Items**
- **Example of Editing Item Attributes**

**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 users’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.

**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 **Application Builder** icon.
2. Select an application.
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.

**About Translating Applications for Multibyte Languages**
If your application must run in several languages simultaneously (such as Chinese and Japanese), consider configuring your database with a character set to support all of the languages. UTF8 and AL32UTF8 are the character sets you can use to support almost all languages around the world.

**Understanding the Translation Process**
This section describes the process of translating an application. To translate an application developed in Application 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**
- **Step 2: Seed and Download to a Translation File**
- **Step 3: Translate the XLIFF File**
- **Step 4: Upload and Apply a Translated XLIFF Document**
- **Step 5: Publish the Application**
- **Manually Editing a Translation**

**See Also:** "Translating Messages" on page 21-13 and "Translating Data That Supports List of Values" on page 21-34

**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 **Application 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" on page 21-7

---

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**
- **Downloading an XLIFF File**

Seeding Translatable Text

Seeding the translation copies all translatable text into the Translation Text repository. After you specify the language and seed the Translation Text, you can then generate and export an XLIFF file for translation.

The seeding process keeps your primary language application synchronized with the Translation Text repository. You should run the seed process any time your primary language application changes.

To seed translatable text:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **Application Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Translate Application**.
The Translate page appears.

2. Click Seed translatable text.
3. Select the appropriate target application.
4. Click Seed.

When completed, the following message displays:
Successfully seeded translatable text.

Downloading an XLIFF File
Once the translation mappings are established the translatable text within the application is seeded into a translation repository. This repository is then exported to an XML Localization Interchange File Format (XLIFF) file for translation.

■ About the XLIIFF Export Page
■ Downloading a XLIFF file for an Application
■ Downloading an XLIFF File for an Application Page
■ About Including XLIFF Target Elements

About the XLIFF Export Page The XLIFF Export page is divided into two sections. Use the upper section to export translatable text for an entire application (that is, all pages, lists of values, messages, and so on). Use the lower section to export translatable text for a specific page.

---

**Note:** XML Localization Interchange File Format (XLIFF) is an XML-based format for exchanging localization data. For information about the XLIFF or to view the XLIFF specification, see:

http://www.xliff.org

---

Downloading a XLIFF file for an Application To download an XLIFF file for a complete application:
2. Navigate to the Translate Application page:
   a. On the Workspace home page, click the Application 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.

**Downloading an XLIFF File for an Application Page** To download an XLIFF file for a specific page:

1. Seed the translatable text as described in "Seeding Translatable Text" on page 21-7.
2. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **Application 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.

**About Including XLIFF Target Elements** When Oracle Application Express generates an XLIFF document, each document contains multiple translation units. Each translation unit consists of a source element and a target element. The XLIFF document can be generated with both the source and target elements for each translation unit. You have the option of generating a file containing only source elements. The updated translations will be applied from the target elements of the translation units.

**Step 3: Translate the XLIFF File**

After you export a translatable file to XLIFF format, you can translate it into the appropriate languages. Because XLIFF is an open standard XML file for exchanging translations, most translation vendors should support it. Oracle Application Express only supports XLIFF files encoded in UTF-8 character sets. In other words, it exports XLIFF files for translation in UTF-8 and assumes that the translated XLIFF files are in the same character set.

Translation is a time-consuming task. Oracle Application Express supports incremental translation so that application development can be done in parallel with the translation. An XLIFF file can be translated and uploaded to Oracle Application Express even when only part of the XLIFF file is translated. For strings that have no translation in the corresponding translated application, Oracle Application Express uses the corresponding ones in the primary language.
Step 4: Upload and Apply a Translated XLIFF Document

After your XLIFF document has been translated, the next step is to upload and then apply it.

- Uploading a Translated XLIFF Document
- Applying an Uploaded XLIFF Document
- Deleting an Uploaded XLIFF Document

Uploading a Translated XLIFF Document
To upload a translated XLIFF document:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
   d. Under Globalization, click Translate Application.
      The Translate page appears.
2. Click Apply XLIFF translation files.
   The XLIFF Translation Files page appears.
3. Click Upload Files.

4. On the XLIFF Upload page:
   a. Locate the file to be uploaded.
   b. Click Upload.
      The uploaded document appears on the XLIFF Translation Files page.

Applying an Uploaded XLIFF Document
After you upload an XLIFF document, the next step is to apply the XLIFF document and then publish the translated application. When you apply an XLIFF document, the Application Express engine parses the file and then updates the translation tables with the new translatable text.

Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternate languages.

Remember that to run an application in an alternative language, you must run it with globalization settings that cause an alternative language version to display. For example, if the language is derived from the browser language, you must set the browser language to the same language as the translated application.

See Also: "Specifying the Primary Language for an Application" on page 21-4

See Also: For more information about the XLIFF, or to view the XLIFF specification, see: http://www.xliff.org
To apply a translated XLIFF document:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the Application 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 "Step 5: Publish the Application" on page 21-11.

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 Application 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" on page 21-4

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 \texttt{Application Builder} icon.
   b. Select an application.
   c. Click \texttt{Shared Components}.
   d. Under Globalization, click \texttt{Translate Application}.

   The Translate Application page appears.

2. Click \texttt{Publish translated applications}.

   The XLIFF Translation Files page appears.

3. Select the items you want to publish and click \texttt{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 Application 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 Application Builder, you must ensure that you have correctly configured the application Globalization attributes.

\section*{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. See "Step 1: Map the Target Language" on page 21-6.
3. Navigate to the Translatable Text:
   a. On the Workspace home page, click the \texttt{Application Builder} icon.
   b. Select an application.
   c. Click \texttt{Shared Components}.
   d. Under Globalization, click \texttt{Translate Application}.
4. Under Translation Utilities, click \texttt{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. To learn more, see "Customizing Interactive Reports as an End User" on page 10-57.
7. To edit translatable text:
   a. Click the \texttt{Edit} icon.

   The Translatable Text page appears.
Translating Messages

This section describes when and how to translate messages in an Oracle Application Express application.

- About Translating Messages
- Translating Messages Used in PL/SQL Procedures
- Translating Messages Used Internally by Oracle Application Express

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.

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
- Defining a Translation Message

See Also: APEX_LANG in Oracle Application Express API Reference

About Translating Messages Used in PL/SQL Procedures

If your application includes PL/SQL regions or PL/SQL processes or calls PL/SQL package, procedures, or functions, you may need to translate generated HTML. First, you define each message on the Translatable Messages page. Second, you use the APEX_LANG.MESSAGE API to translate the messages from PL/SQL stored procedures, functions, triggers, or packaged procedures and functions.

Defining a Translation Message

You create translatable messages on the Translate Messages page.

To define a new translation message:

1. Navigate to the Translate Application page:
   - On the Workspace home page, click the Application Builder icon.
   - Select an application.
c. Click Shared Components.

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.

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
- Editing Text Messages
- Internal Messages Requiring Translation
- Interactive Report Messages Requiring Translation

Creating Text Messages

To create a translation message:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.

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. See "Internal Messages Requiring Translation" on page 21-15 and "Interactive Report Messages Requiring Translation" on page 21-17.
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.

Editing Text Messages
To edit text messages:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.

2. On edit a text message:
   a. Click the message name.
   b. Make the appropriate edits.
      If the English text message contains positional substitution values (for example, %0, %1), ensure that your defined message also contains the same named and number of positional substitution values.
   c. Click Apply Changes.

3. To access a Grid Edit page:
   a. Click Grid Edit.
      The Grid Edit page appears.
   b. Make the appropriate edits.
      If the English text message contains positional substitution values (for example, %0, %1), ensure that your defined message also contains the same named and number of positional substitution values.
   c. Click Apply Changes

Tip: To add a new message from the Grid Edit page, click Add Text Message at the bottom of the page.

Internal Messages Requiring Translation
Table 21–2 lists the internal messages that require translation.

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEX.AUTHENTICATION.LOGIN_THROTTLE.COUNTER</td>
<td>Please wait &lt;span id=&quot;apex_login_throttle_sec&quot;&gt;%0&lt;/span&gt; seconds to login again.</td>
</tr>
<tr>
<td>APEX.AUTHENTICATION.LOGIN_THROTTLE.ERROR</td>
<td>The login attempt has been blocked.</td>
</tr>
<tr>
<td>APEX.DATA_HAS_CHANGED</td>
<td>Current version of data in database has changed since user initiated update process. current checksum = &quot;%0&quot; application checksum = &quot;%1&quot;.</td>
</tr>
</tbody>
</table>
### Table 21–2 (Cont.) Internal Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEX.DATEPICKER.ICON_TEXT</td>
<td>Popup Calendar: %0</td>
</tr>
<tr>
<td>APEX.DATEPICKER.VALUE_GREATER_MAX_DATE</td>
<td>#LABEL# is greater than specified maximum date %0.</td>
</tr>
<tr>
<td>APEX.DATEPICKER.VALUE_INVALID</td>
<td>#LABEL# does not match format %0.</td>
</tr>
<tr>
<td>APEX.DATEPICKER.VALUE_LESS_MIN_DATE</td>
<td>#LABEL# is less than specified minimum date %0.</td>
</tr>
<tr>
<td>APEX.DATEPICKER.VALUE_NOT_BETWEEN_MIN_MAX</td>
<td>#LABEL# is not between the valid range of %0 and %1.</td>
</tr>
<tr>
<td>APEX.DATEPICKER.VALUE_NOT_IN_YEAR_RANGE</td>
<td>#LABEL# is not within valid year range of %0 and %1.</td>
</tr>
<tr>
<td>APEX.FILE_BROWSE.DOWNLOAD_LINK_TEXT</td>
<td>Download</td>
</tr>
<tr>
<td>APEX.GO_TO_ERROR</td>
<td>Go to error</td>
</tr>
<tr>
<td>APEX.ITEM_TYPE.SLIDER.VALUE_NOT_BETWEEN_MIN_MAX</td>
<td>#LABEL# is not between the valid range of %0 and %1.</td>
</tr>
<tr>
<td>APEX.ITEM_TYPE.SLIDER.VALUE_NOT_MULTIPLE_OF_STEP</td>
<td>#LABEL# is not a multiple of %0.</td>
</tr>
<tr>
<td>APEX.ITEM_TYPE.YES_NO.INVALID_VALUE</td>
<td>#LABEL# must match to the values %0 and %1.</td>
</tr>
<tr>
<td>APEX.ITEM_TYPE.YES_NO.NO_LABEL</td>
<td>No</td>
</tr>
<tr>
<td>APEX.ITEM_TYPE.YES_NO.YES_LABEL</td>
<td>Yes</td>
</tr>
<tr>
<td>APEX.NUMBER_FIELD.VALUE_GREATER_MAX_VALUE</td>
<td>#LABEL# is greater than specified maximum %0.</td>
</tr>
<tr>
<td>APEX.NUMBER_FIELD.VALUE_INVALID</td>
<td>#LABEL# must be Numeric.</td>
</tr>
<tr>
<td>APEX.NUMBER_FIELD.VALUE_INVALID2</td>
<td>#LABEL# does not match number format %0 (For example, %1).</td>
</tr>
<tr>
<td>APEX.NUMBER_FIELD.VALUE_LESS_MIN_VALUE</td>
<td>#LABEL# is less than specified minimum %0.</td>
</tr>
<tr>
<td>APEX.NUMBER_FIELD.VALUE_NOT_BETWEEN_MIN_MAX</td>
<td>#LABEL# is not between the valid range of %0 and %1.</td>
</tr>
<tr>
<td>APEX.POPUP_LOV.ICON_TEXT</td>
<td>Popup List of Values: %0.</td>
</tr>
<tr>
<td>APEX.PAGINATION.NEXT</td>
<td>Next</td>
</tr>
<tr>
<td>APEX.PAGINATION.NEXT_SET</td>
<td>Next Set</td>
</tr>
<tr>
<td>APEX.PAGINATION.PREVIOUS</td>
<td>Previous</td>
</tr>
<tr>
<td>APEX.PAGINATION.PREVIOUS_SET</td>
<td>Previous Set</td>
</tr>
<tr>
<td>APEX.PAGINATION.PREVIOUS_SET</td>
<td>Previous Set</td>
</tr>
<tr>
<td>APEX.PAGINATION.PREVIOUS_SET</td>
<td>Previous Set</td>
</tr>
<tr>
<td>APEX.REPORT_TOTAL</td>
<td>Report total</td>
</tr>
<tr>
<td>APEX.RESET</td>
<td>Reset pagination</td>
</tr>
<tr>
<td>APEX.SINCE_DAYS_AGO</td>
<td>%0 days ago</td>
</tr>
<tr>
<td>APEX.SINCE_DAYS_FROM_NOW</td>
<td>%0 days from now</td>
</tr>
<tr>
<td>APEX.SINCE_HOURS_AGO</td>
<td>%0 hours ago</td>
</tr>
<tr>
<td>APEX.SINCE_HOURS_FROM_NOW</td>
<td>%0 hours from now</td>
</tr>
</tbody>
</table>
Table 21–2 (Cont.) Internal Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINCE_MINUTES_AGO</td>
<td>%0 minutes ago</td>
</tr>
<tr>
<td>SINCE_MINUTES_FROM_NOW</td>
<td>%0 minutes from now</td>
</tr>
<tr>
<td>SINCE_MONTHS_AGO</td>
<td>%0 months ago</td>
</tr>
<tr>
<td>SINCE_MONTHS_FROM_NOW</td>
<td>%0 months from now</td>
</tr>
<tr>
<td>SINCE_NOW</td>
<td>Now</td>
</tr>
<tr>
<td>SINCE_SECONDS_AGO</td>
<td>%0 seconds ago</td>
</tr>
<tr>
<td>SINCE_SECONDS_FROM_NOW</td>
<td>%0 seconds from now</td>
</tr>
<tr>
<td>SINCE_WEEKS_AGO</td>
<td>%0 weeks ago</td>
</tr>
<tr>
<td>SINCE_WEEKS_FROM_NOW</td>
<td>%0 weeks from now</td>
</tr>
<tr>
<td>SINCE_YEARS_AGO</td>
<td>%0 years ago</td>
</tr>
<tr>
<td>SINCE_YEARS_FROM_NOW</td>
<td>%0 years from now</td>
</tr>
<tr>
<td>TOTAL</td>
<td>Total</td>
</tr>
<tr>
<td>WWV_FLOW_UTILITIES.CAL</td>
<td>Calendar</td>
</tr>
<tr>
<td>WWV_FLOW_UTILITIES.CLOSE</td>
<td>Close</td>
</tr>
<tr>
<td>WWV_FLOW_UTILITIES.OK</td>
<td>Ok</td>
</tr>
<tr>
<td>WWV_RENDER_REPORT3.FOUND_BUT_NOT_DISPLAYED</td>
<td>Minimum row requested: %0, rows found but not displayed: %1</td>
</tr>
<tr>
<td>WWV_RENDER_REPORT3.SORT_BY_THIS_COLUMN</td>
<td>Sort by this column.</td>
</tr>
<tr>
<td>WWV_RENDER_REPORT3.UNSAVED_DATA</td>
<td>This form contains unsaved changes. Press Ok to proceed without</td>
</tr>
<tr>
<td></td>
<td>saving your changes.</td>
</tr>
<tr>
<td>WWV_RENDER_REPORT3.X_Y_OF_MORE_THAN_Z</td>
<td>row(s) %0 - %1 of more than %2</td>
</tr>
<tr>
<td>WWV_RENDER_REPORT3.X_Y_OF_Z</td>
<td>row(s) %0 - %1 of %2</td>
</tr>
<tr>
<td>WWV_RENDER_REPORT3.X_Y_OF_Z_2</td>
<td>%0 - %1 of %2</td>
</tr>
</tbody>
</table>

Interactive Report Messages Requiring Translation

Table 21–3 lists the interactive report messages that require translation.

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>4150_COLUMN_NUMBER</td>
<td>Column %0</td>
</tr>
<tr>
<td>APEXIR_3D</td>
<td>3D</td>
</tr>
<tr>
<td>APEXIR_ACTIONS</td>
<td>Actions</td>
</tr>
<tr>
<td>APEXIR_ACTIONS_MENU</td>
<td>Actions Menu</td>
</tr>
<tr>
<td>APEXIR_ADD</td>
<td>Add</td>
</tr>
<tr>
<td>APEXIR_ADD_FUNCTION</td>
<td>Add Function</td>
</tr>
<tr>
<td>APEXIR_ADD_GROUP_BY_COLUMN</td>
<td>Add Group By Column</td>
</tr>
<tr>
<td>APEXIR_ADD_PIVOT_COLUMN</td>
<td>Add Pivot Column</td>
</tr>
<tr>
<td>APEXIR_ADD_ROW_COLUMN</td>
<td>Add Row Column</td>
</tr>
<tr>
<td>APEXIR_ADD_SUBSCRIPTION</td>
<td>Add Subscription</td>
</tr>
</tbody>
</table>
### Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_AGG_AVG</td>
<td>Average</td>
</tr>
<tr>
<td>APEXIR_AGG_COUNT</td>
<td>Count</td>
</tr>
<tr>
<td>APEXIR_AGG_MAX</td>
<td>Maximum</td>
</tr>
<tr>
<td>APEXIR_AGG_MEDIAN</td>
<td>Median</td>
</tr>
<tr>
<td>APEXIR_AGG_MIN</td>
<td>Minimum</td>
</tr>
<tr>
<td>APEXIR_AGG_MODE</td>
<td>Mode</td>
</tr>
<tr>
<td>APEXIR_AGG_SUM</td>
<td>Sum</td>
</tr>
<tr>
<td>APEXIR_AGGREGATE</td>
<td>Aggregate</td>
</tr>
<tr>
<td>APEXIR_AGGREGATE_DESCRIPTION</td>
<td>Aggregates are displayed after each control break and at the end of the report.</td>
</tr>
<tr>
<td>APEXIR_AGGREGATION</td>
<td>Aggregation</td>
</tr>
<tr>
<td>APEXIR_ALL</td>
<td>All</td>
</tr>
<tr>
<td>APEXIR_ALL_COLUMNS</td>
<td>All Columns</td>
</tr>
<tr>
<td>APEXIR_ALL_ROWS</td>
<td>All Rows</td>
</tr>
<tr>
<td>APEXIR_ALTERNATIVE</td>
<td>Alternative</td>
</tr>
<tr>
<td>APEXIR_ALTERNATIVE_DEFAULT_NAME</td>
<td>Alternative Default: %0</td>
</tr>
<tr>
<td>APEXIR_AND</td>
<td>and</td>
</tr>
<tr>
<td>APEXIR_APPLY</td>
<td>Apply</td>
</tr>
<tr>
<td>APEXIR_AS_OF</td>
<td>As of %0</td>
</tr>
<tr>
<td>APEXIR_ASCENDING</td>
<td>Ascending</td>
</tr>
<tr>
<td>APEXIR_AVERAGE_X</td>
<td>Average %0</td>
</tr>
<tr>
<td>APEXIR_BETWEEN</td>
<td>between</td>
</tr>
<tr>
<td>APEXIR_BGCOLOR</td>
<td>Background Color</td>
</tr>
<tr>
<td>APEXIR_BLUE</td>
<td>blue</td>
</tr>
<tr>
<td>APEXIR_BOTTOM</td>
<td>Bottom</td>
</tr>
<tr>
<td>APEXIRCALENDAR</td>
<td>Calendar</td>
</tr>
<tr>
<td>APEXIR_CANCEL</td>
<td>Cancel</td>
</tr>
<tr>
<td>APEXIR_CATEGORY</td>
<td>Category</td>
</tr>
<tr>
<td>APEXIR_CELL</td>
<td>Cell</td>
</tr>
<tr>
<td>APEXIR_CHART</td>
<td>Chart</td>
</tr>
<tr>
<td>APEXIR_CHART_INITIALIZING</td>
<td>Initializing...</td>
</tr>
<tr>
<td>APEXIR_CHART_LABEL_NOT_NULL</td>
<td>Chart label must be specified</td>
</tr>
<tr>
<td>APEXIR_CHART_MAX_ROW_CNT</td>
<td>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.</td>
</tr>
<tr>
<td>APEXIR_CHART_TYPE</td>
<td>Chart Type</td>
</tr>
<tr>
<td>APEXIR_CHECK_ALL</td>
<td>Check All</td>
</tr>
</tbody>
</table>
### Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_CHOOSE_DOWNLOAD_FORMAT</td>
<td>Choose report download format</td>
</tr>
<tr>
<td>APEXIR_CLEAR</td>
<td>clear</td>
</tr>
<tr>
<td>APEXIR_COLUMN</td>
<td>Column</td>
</tr>
<tr>
<td>APEXIR_COLUMN_FILTER</td>
<td>Filter...</td>
</tr>
<tr>
<td>APEXIR_COLUMN_HEADING</td>
<td>Column Heading</td>
</tr>
<tr>
<td>APEXIR_COLUMN_HEADING_MENU</td>
<td>Column Heading Menu</td>
</tr>
<tr>
<td>APEXIR_COLUMN_INFO</td>
<td>Column Information</td>
</tr>
<tr>
<td>APEXIR_COLUMN_N</td>
<td>Column %0</td>
</tr>
<tr>
<td>APEXIR_COLUMNS</td>
<td>Columns</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_CONTAINS</td>
<td>contains</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_DOESNOT_CONTAIN</td>
<td>does not contain</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_IN</td>
<td>in</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_IS_IN_LAST</td>
<td>is in the last</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_IS_IN_NEXT</td>
<td>is in the next</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_IS_NOT_NULL</td>
<td>is not null</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_IS_NULL</td>
<td>is null</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_IS_NOT_IN_LAST</td>
<td>is not in the last</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_IS_NOT_IN_NEXT</td>
<td>is not in the next</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_LIKE</td>
<td>like</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_NOT_IN</td>
<td>not in</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_NOT_LIKE</td>
<td>not like</td>
</tr>
<tr>
<td>APEXIR_COMPARISON_REGEXP_LIKE</td>
<td>matches regular expression</td>
</tr>
<tr>
<td>APEXIR_COMPUTATION</td>
<td>Computation</td>
</tr>
<tr>
<td>APEXIR_COMPUTATION_EXPRESSION</td>
<td>Computation Expression</td>
</tr>
<tr>
<td>APEXIR_COMPUTATION_FOOTER</td>
<td>Create a computation using column aliases.</td>
</tr>
<tr>
<td>APEXIR_COMPUTATION_FOOTER_E1</td>
<td>(B+C)*100</td>
</tr>
<tr>
<td>APEXIR_COMPUTATION_FOOTER_E2</td>
<td>INITCAP(B)</td>
</tr>
<tr>
<td>APEXIR_COMPUTATION_FOOTER_E3</td>
<td>CASE WHEN A = 10 THEN B + C ELSE B END</td>
</tr>
<tr>
<td>APEXIR_COMPUTE</td>
<td>Compute</td>
</tr>
<tr>
<td>APEXIR_CONTROL_BREAK</td>
<td>Control Break</td>
</tr>
<tr>
<td>APEXIR_CONTROL_BREAK_COLUMNS</td>
<td>Control Break Columns</td>
</tr>
<tr>
<td>APEXIR_CONTROL_BREAKS</td>
<td>Control Breaks</td>
</tr>
<tr>
<td>APEXIR_COUNT_DISTINCT</td>
<td>Count Distinct</td>
</tr>
<tr>
<td>APEXIR_COUNT_DISTINCT_X</td>
<td>Count Distinct</td>
</tr>
<tr>
<td>APEXIR_COUNT_X</td>
<td>Count %0</td>
</tr>
<tr>
<td>APEXIR_DAILY</td>
<td>Daily</td>
</tr>
<tr>
<td>APEXIR_DATA_AS_OF</td>
<td>Report data as of %0 minutes ago.</td>
</tr>
</tbody>
</table>
### Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_DATE</td>
<td>Date</td>
</tr>
<tr>
<td>APEXIR_DAY</td>
<td>Day</td>
</tr>
<tr>
<td>APEXIR_DEFAULT</td>
<td>Default</td>
</tr>
<tr>
<td>APEXIR_DEFAULT_REPORT_TYPE</td>
<td>Default Report Type</td>
</tr>
<tr>
<td>APEXIR_DELETE</td>
<td>Delete</td>
</tr>
<tr>
<td>APEXIR_DELETE_CHECKED</td>
<td>Delete Checked</td>
</tr>
<tr>
<td>APEXIR_DELETE_CONFIRM</td>
<td>Would you like to delete these report settings?</td>
</tr>
<tr>
<td>APEXIR_DELETE_CONFIRM_JS_DIALOG</td>
<td>Would you like to perform this delete action?</td>
</tr>
<tr>
<td>APEXIR_DELETE_DEFAULT_REPORT</td>
<td>Delete Default Report</td>
</tr>
<tr>
<td>APEXIR_DELETE_REPORT</td>
<td>Delete Report</td>
</tr>
<tr>
<td>APEXIR_DESCENDING</td>
<td>Descending</td>
</tr>
<tr>
<td>APEXIR_DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>APEXIR_DETAIL_VIEW</td>
<td>Single Row View</td>
</tr>
<tr>
<td>APEXIR_DIRECTION</td>
<td>Direction</td>
</tr>
<tr>
<td>APEXIR_DISABLE</td>
<td>Disable</td>
</tr>
<tr>
<td>APEXIR_DISABLED</td>
<td>Disabled</td>
</tr>
<tr>
<td>APEXIR_DISPLAY</td>
<td>Display</td>
</tr>
<tr>
<td>APEXIR_DISPLAY_IN_REPORT</td>
<td>Display in Report</td>
</tr>
<tr>
<td>APEXIR_DISPLAYED</td>
<td>Displayed</td>
</tr>
<tr>
<td>APEXIR_DISPLAYED_COLUMNS</td>
<td>Displayed Columns</td>
</tr>
<tr>
<td>APEXIR_DO_NOT_AGGREGATE</td>
<td>- Do not aggregate -</td>
</tr>
<tr>
<td>APEXIR_DO_NOT_DISPLAY</td>
<td>Do Not Display</td>
</tr>
<tr>
<td>APEXIR_DOWN</td>
<td>Down</td>
</tr>
<tr>
<td>APEXIR_DOWNLOAD</td>
<td>Download</td>
</tr>
<tr>
<td>APEXIR_DUPLICATE_PIVOT_COLUMN</td>
<td>Duplicate pivot column. Pivot column list must be unique.</td>
</tr>
<tr>
<td>APEXIR_EDIT</td>
<td>Edit</td>
</tr>
<tr>
<td>APEXIR_EDIT_ALTERNATIVE_DEFAULT</td>
<td>Edit Alternative Default</td>
</tr>
<tr>
<td>APEXIR_EDIT_CHART</td>
<td>Edit Chart Settings</td>
</tr>
<tr>
<td>APEXIR_EDIT_CHART2</td>
<td>Edit Chart</td>
</tr>
<tr>
<td>APEXIR_EDIT_FILTER</td>
<td>Edit Filter</td>
</tr>
<tr>
<td>APEXIR_EDIT_GROUP_BY</td>
<td>Edit Group By</td>
</tr>
<tr>
<td>APEXIR_EDIT_HIGHLIGHT</td>
<td>Edit Highlight</td>
</tr>
<tr>
<td>APEXIR_EDIT_PIVOT</td>
<td>Edit Pivot</td>
</tr>
<tr>
<td>APEXIR_EMAIL</td>
<td>Email</td>
</tr>
<tr>
<td>APEXIR_EMAIL_ADDRESS</td>
<td>Email Address</td>
</tr>
<tr>
<td>APEXIR_EMAIL_BCC</td>
<td>Bcc</td>
</tr>
</tbody>
</table>
### Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_EMAIL_BODY</td>
<td>Body</td>
</tr>
<tr>
<td>APEXIR_EMAIL_CC</td>
<td>Cc</td>
</tr>
<tr>
<td>APEXIR_EMAIL_FREQUENCY</td>
<td>Frequency</td>
</tr>
<tr>
<td>APEXIR_EMAIL_NOT_CONFIGURED</td>
<td>Email has not been configured for this application. Please contact your administrator.</td>
</tr>
<tr>
<td>APEXIR_EMAIL_REQUIRED</td>
<td>Email Address must be specified.</td>
</tr>
<tr>
<td>APEXIR_EMAILSEE_ATTACHED</td>
<td>See attached.</td>
</tr>
<tr>
<td>APEXIR_EMAIL_SUBJECT</td>
<td>Subject</td>
</tr>
<tr>
<td>APEXIR_EMAIL_SUBJECT_REQUIRED</td>
<td>Email Subject must be specified.</td>
</tr>
<tr>
<td>APEXIR_EMAIL_TO</td>
<td>To</td>
</tr>
<tr>
<td>APEXIR_ENABLE</td>
<td>Enable</td>
</tr>
<tr>
<td>APEXIR_ENABLE_DISABLE_ALT</td>
<td>Enable/Disable</td>
</tr>
<tr>
<td>APEXIR_ENABLED</td>
<td>Enabled</td>
</tr>
<tr>
<td>APEXIR_ERROR</td>
<td>Error</td>
</tr>
<tr>
<td>APEXIR_EXAMPLES</td>
<td>Examples</td>
</tr>
<tr>
<td>APEXIR_EXAMPLES_WITH_COLON</td>
<td>Examples:</td>
</tr>
<tr>
<td>APEXIR_EXCLUDE_NULL</td>
<td>Exclude Null Values</td>
</tr>
<tr>
<td>APEXIR_EXPAND_COLLAPSE_ALT</td>
<td>Expand/Collapse</td>
</tr>
<tr>
<td>APEXIR_EXPRESSION</td>
<td>Expression</td>
</tr>
<tr>
<td>APEXIR_FILTER</td>
<td>Filter</td>
</tr>
<tr>
<td>APEXIR_FILTER_EXPRESSION</td>
<td>Filter Expression</td>
</tr>
<tr>
<td>APEXIR_FILTER_TYPE</td>
<td>Filter Type</td>
</tr>
<tr>
<td>APEXIR_FILTERS</td>
<td>Filters</td>
</tr>
<tr>
<td>APEXIR_FINDER_ALT</td>
<td>Select columns to search.</td>
</tr>
<tr>
<td>APEXIR_FLASHBACK</td>
<td>Flashback</td>
</tr>
<tr>
<td>APEXIR_FLASHBACK_DESCRIPTION</td>
<td>A flashback query enables you to view the data as it existed at a previous point in time.</td>
</tr>
<tr>
<td>APEXIR_FLASHBACK_ERROR_MSG</td>
<td>Unable to perform flashback request.</td>
</tr>
<tr>
<td>APEXIR_FORMAT</td>
<td>Format</td>
</tr>
<tr>
<td>APEXIR_FORMAT_MASK</td>
<td>Format Mask</td>
</tr>
<tr>
<td>APEXIR_FUNCTION</td>
<td>Function</td>
</tr>
<tr>
<td>APEXIR_FUNCTION_N</td>
<td>Function %0</td>
</tr>
<tr>
<td>APEXIR_FUNCTIONS</td>
<td>Functions</td>
</tr>
<tr>
<td>APEXIR_FUNCTIONS_OPERATORS</td>
<td>Functions / Operators</td>
</tr>
<tr>
<td>APEXIR_GO</td>
<td>Go</td>
</tr>
<tr>
<td>APEXIR_GREEN</td>
<td>green</td>
</tr>
<tr>
<td>APEXIR_GROUP_BY</td>
<td>Group By</td>
</tr>
<tr>
<td>APEXIR_GROUP_BY_COL_NOT_NULL</td>
<td>Group by column must be specified.</td>
</tr>
</tbody>
</table>
Table 21–3  (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_GROUP_BY_COLUMN</td>
<td>Group By Column</td>
</tr>
<tr>
<td>APEXIR_GROUP_BY_MAX_ROW_CNT</td>
<td>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.</td>
</tr>
<tr>
<td>APEXIR_GROUP_BY_SORT</td>
<td>Group By Sort</td>
</tr>
<tr>
<td>APEXIR_GROUP_BY_SORT_ORDER</td>
<td>Group By Sort Order</td>
</tr>
<tr>
<td>APEXIR_HCOLUMN</td>
<td>Horizontal Column</td>
</tr>
<tr>
<td>APEXIR_HELP</td>
<td>Help</td>
</tr>
</tbody>
</table>
| APEXIR_HELP_01                   | 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/>
An Interactive Report can be customized in three ways: the search bar, actions menu and column heading menu. |
| APEXIR_HELP_ACTIONS_MENU         | The actions menu is used to customize the display of your Interactive Report. |
| APEXIR_HELP_AGGREGATE            | 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/>
<ul><li><b>Aggregation</b> allows you to select a previously defined aggregation to edit.</li><li><b>Function</b> is the function to be performed (e.g. SUM, MIN).</li><li><b>Column</b> is used to select the column to apply the mathematical function to. Only numeric columns will be displayed.</li></ul>
### Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_HELP_CHART</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>&lt;p&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;ul&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Chart Type&lt;/b&gt; identifies the chart type to include. Select from horizontal bar, vertical bar, pie or line.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Label&lt;/b&gt; allows you to select the column to be used as the label.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Axis Title for Label&lt;/b&gt; is the title that will display on the axis associated with the column selected for Label. This is not available for pie chart.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Value&lt;/b&gt; 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.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Axis Title for Value&lt;/b&gt; is the title that will display on the axis associated with the column selected for Value. This is not available for pie chart.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Function&lt;/b&gt; is an optional function to be performed on the column selected for Value.</td>
</tr>
<tr>
<td></td>
<td>&lt;/ul&gt;</td>
</tr>
<tr>
<td></td>
<td>Clicking on any column heading exposes a column heading menu.</td>
</tr>
<tr>
<td></td>
<td>&lt;p&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;ul&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Sort Ascending&lt;/b&gt; sorts the report by the column in ascending order.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Sort Descending&lt;/b&gt; sorts the report by the column in descending order.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Hide Column&lt;/b&gt; hides the column.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Break Column&lt;/b&gt; creates a break group on the column. This pulls the column out of the report as a master record.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Column Information&lt;/b&gt; displays help text about the column, if available.</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;Text Area&lt;/b&gt; 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').</td>
</tr>
<tr>
<td></td>
<td>&lt;li&gt;&lt;b&gt;List of Unique Values&lt;/b&gt; 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').</td>
</tr>
</tbody>
</table>

Computations allow you to add computed columns to your report. These can be mathematical computations (e.g. \(\text{NBR\_HOURS}/24\)) or standard Oracle functions applied to existing columns (some have been displayed for example, others, like TO\_DATE, can also be used).

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. S99999).

Computation is the computation to be performed. Within the computation, columns are referenced using the aliases displayed.

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.

An example computation to display Total Compensation is:

\[
\text{CASE WHEN A = 'SALES' THEN B + C ELSE B END}
\]

(where A is ORGANIZATION, B is SALARY and C is COMMISSION)

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.

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.

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 21-3  (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_HELP_FILTER</td>
<td>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%).</td>
</tr>
<tr>
<td>APEXIR_HELP_FLASHBACK</td>
<td>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.</td>
</tr>
<tr>
<td>APEXIR_HELP_GROUP_BY</td>
<td>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: &lt;p&gt;&lt;/p&gt; &lt;ul&gt; &lt;li&gt;the columns on which to group&lt;/li&gt; &lt;li&gt;the columns to aggregate along with the function to be performed (average, sum, count, etc.)&lt;/li&gt; &lt;/ul&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_HIGHLIGHT</td>
<td>Highlighting allows you to define a filter. The rows that meet the filter are highlighted using the characteristics associated with the filter. &lt;p&gt;&lt;/p&gt; &lt;ul&gt; &lt;li&gt;Name is used only for display.&lt;/li&gt; &lt;li&gt;Sequence identifies the sequence in which the rules will be evaluated.&lt;/li&gt; &lt;li&gt;Enabled identifies if the rule is enabled or disabled.&lt;/li&gt; &lt;li&gt;Highlight Type identifies whether the Row or Cell should be highlighted. If Cell is selected, the column referenced in the Highlight Condition is highlighted.&lt;/li&gt; &lt;li&gt;Background Color is the new color for the background of the highlighted area.&lt;/li&gt; &lt;li&gt;Text Color is the new color for the text in the highlighted area.&lt;/li&gt; &lt;li&gt;Highlight Condition defines your filter condition.&lt;/li&gt; &lt;/ul&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_PIVOT</td>
<td>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: &lt;p&gt;&lt;/p&gt; &lt;ul&gt; &lt;li&gt;the columns on which to pivot&lt;/li&gt; &lt;li&gt;the columns to display as rows&lt;/li&gt; &lt;li&gt;the columns to aggregate along with the function to be performed (average, sum, count, etc.)&lt;/li&gt; &lt;/ul&gt;</td>
</tr>
</tbody>
</table>
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.

For each report setting, you can:
- **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.

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.

**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**
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:

- **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.

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).
At the top of each report page is a search region. The region provides the following features:

- **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).

Please note that all features may not be available for each report.

**Actions Menu** enables you to customize a report. See the sections that follow.

**Select columns icon** enables you to identify which column to search (or all).

**Reports** displays alternate default and saved private or public reports.

**Rows** sets the number of records to display per page.

**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.

**View Icons** switches between the icon, report, detail, chart, group by, and pivot views of the report if they are defined.

**Select columns** is 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 **.*

**Sort** is 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.

**Subscription** is used 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.

**Hide Column**

**Highlight**

### Table 21-3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_HELP_SEARCH_BAR</td>
<td>At the top of each report page is a search region. The region provides the following features:</td>
</tr>
<tr>
<td>APEXIR_HELP_SEARCH_BAR_ACTIONS_MENU</td>
<td>&lt;li&gt;&lt;b&gt;Actions Menu&lt;/b&gt; enables you to customize a report. See the sections that follow.&lt;/li&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_SEARCH_BAR_FINDER</td>
<td>&lt;li&gt;&lt;b&gt;Select columns icon&lt;/b&gt; enables you to identify which column to search (or all).&lt;/li&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_SEARCH_BAR_REPORTS</td>
<td>&lt;li&gt;&lt;b&gt;Reports&lt;/b&gt; displays alternate default and saved private or public reports.&lt;/li&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_SEARCH_BAR_ROWS</td>
<td>&lt;li&gt;&lt;b&gt;Rows&lt;/b&gt; sets the number of records to display per page.&lt;/li&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_SEARCH_BAR_TEXTBOX</td>
<td>&lt;li&gt;&lt;b&gt;Text area&lt;/b&gt; enables you to enter case insensitive search criteria (wild card characters are implied).&lt;/li&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_SEARCH_BAR_VIEW</td>
<td>&lt;li&gt;&lt;b&gt;View Icons&lt;/b&gt; switches between the icon, report, detail, chart, group by, and pivot views of the report if they are defined.&lt;/li&gt;</td>
</tr>
<tr>
<td>APEXIR_HELP_SELECT_COLUMNS</td>
<td>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 &lt;b&gt;**&lt;/b&gt;.</td>
</tr>
<tr>
<td>APEXIR_HELP_SORT</td>
<td>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.</td>
</tr>
<tr>
<td>APEXIR_HELP_SUBSCRIPTION</td>
<td>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.</td>
</tr>
<tr>
<td>APEXIR_HIDE_COLUMN</td>
<td>Hide Column</td>
</tr>
<tr>
<td>APEXIR_HIGHLIGHT</td>
<td>Highlight</td>
</tr>
<tr>
<td>Message Name</td>
<td>English Text</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>APEXIR_HIGHLIGHT_CONDITION</td>
<td>Highlight Condition</td>
</tr>
<tr>
<td>APEXIR_HIGHLIGHT_TYPE</td>
<td>Highlight Type</td>
</tr>
<tr>
<td>APEXIR_HIGHLIGHT_WHEN</td>
<td>Highlight When</td>
</tr>
<tr>
<td>APEXIR_HIGHLIGHTS</td>
<td>Highlights</td>
</tr>
<tr>
<td>APEXIR_INACTIVE_SETTING</td>
<td>1 inactive setting</td>
</tr>
<tr>
<td>APEXIR_INACTIVE_SETTINGS</td>
<td>%0 inactive settings</td>
</tr>
<tr>
<td>APEXIR_INTERACTIVE_REPORT_HELP</td>
<td>Interactive Report Help</td>
</tr>
<tr>
<td>APEXIR_INVALID</td>
<td>Invalid</td>
</tr>
<tr>
<td>APEXIR_INVALID_COMPUTATION</td>
<td>Invalid computation expression. %0</td>
</tr>
<tr>
<td>APEXIR_INVALID_END_DATE</td>
<td>The end date must be greater than the start date.</td>
</tr>
<tr>
<td>APEXIR_INVALID_FILTER_QUERY</td>
<td>Invalid filter query</td>
</tr>
<tr>
<td>APEXIR_INVALID_FILTER</td>
<td>Invalid filter expression. %0</td>
</tr>
<tr>
<td>APEXIR_INVALID_SETTING</td>
<td>1 invalid setting</td>
</tr>
<tr>
<td>APEXIR_INVALID_SETTINGS</td>
<td>%0 invalid settings</td>
</tr>
<tr>
<td>APEXIR_IS_IN_THE_LAST</td>
<td>%0 is in the last %1</td>
</tr>
<tr>
<td>APEXIR_IS_IN_THE_NEXT</td>
<td>%0 is in the next %1</td>
</tr>
<tr>
<td>APEXIR_IS_NOT_IN_THE_LAST</td>
<td>%0 is not in the last %1</td>
</tr>
<tr>
<td>APEXIR_IS_NOT_IN_THE_NEXT</td>
<td>%0 is not in the next %1</td>
</tr>
<tr>
<td>APEXIR_KEYPAD</td>
<td>Keypad</td>
</tr>
<tr>
<td>APEXIR_LABEL</td>
<td>Label</td>
</tr>
<tr>
<td>APEXIR_LABEL_AXIS_TITLE</td>
<td>Axis Title for Label</td>
</tr>
<tr>
<td>APEXIR_LABEL_PREFIX</td>
<td>Label Prefix</td>
</tr>
<tr>
<td>APEXIR_LAST_DAY</td>
<td>Last Day</td>
</tr>
<tr>
<td>APEXIR_LAST_HOUR</td>
<td>Last Hour</td>
</tr>
<tr>
<td>APEXIR_LAST_MONTH</td>
<td>Last Month</td>
</tr>
<tr>
<td>APEXIR_LAST_WEEK</td>
<td>Last Week</td>
</tr>
<tr>
<td>APEXIR_LAST_X_DAYS</td>
<td>Last %0 Days</td>
</tr>
<tr>
<td>APEXIR_LAST_X_HOURS</td>
<td>Last %0 Hours</td>
</tr>
<tr>
<td>APEXIR_LAST_X_YEARS</td>
<td>Last %0 Years</td>
</tr>
<tr>
<td>APEXIR_LAST_YEAR</td>
<td>Last Year</td>
</tr>
<tr>
<td>APEXIR_LINE</td>
<td>Line</td>
</tr>
<tr>
<td>APEXIR_MAX_QUERY_COST</td>
<td>The query is estimated to exceed the maximum allowed resources. Please modify your report settings and try again.</td>
</tr>
<tr>
<td>APEXIR_MAX_ROW_CNT</td>
<td>This query returns more then %0 rows, please filter your data to ensure complete results.</td>
</tr>
<tr>
<td>APEXIR_MAX_X</td>
<td>Maximum %0</td>
</tr>
<tr>
<td>Message Name</td>
<td>English Text</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>APEXIR_MEDIAN_X</td>
<td>Median %0</td>
</tr>
<tr>
<td>APEXIR_MIN_AGO</td>
<td>%0 minutes ago</td>
</tr>
<tr>
<td>APEXIR_MIN_X</td>
<td>Minimum %0</td>
</tr>
<tr>
<td>APEXIR_MONTH</td>
<td>Month</td>
</tr>
<tr>
<td>APEXIR_MONTHLY</td>
<td>Monthly</td>
</tr>
<tr>
<td>APEXIR_MORE_DATA</td>
<td>More Data</td>
</tr>
<tr>
<td>APEXIR_MOVE</td>
<td>Move</td>
</tr>
<tr>
<td>APEXIR_MOVE_ALL</td>
<td>Move All</td>
</tr>
<tr>
<td>APEXIR_NAME</td>
<td>Name</td>
</tr>
<tr>
<td>APEXIR_NEW_AGGREGATION</td>
<td>New Aggregation</td>
</tr>
<tr>
<td>APEXIR_NEW_CATEGORY</td>
<td>- New Category</td>
</tr>
<tr>
<td>APEXIR_NEW_COMPUTATION</td>
<td>New Computation</td>
</tr>
<tr>
<td>APEXIR_NEXT</td>
<td>&gt;</td>
</tr>
<tr>
<td>APEXIR_NEXT_DAY</td>
<td>Next Day</td>
</tr>
<tr>
<td>APEXIR_NEXT_HOUR</td>
<td>Next Hour</td>
</tr>
<tr>
<td>APEXIR_NEXT_MONTH</td>
<td>Next Month</td>
</tr>
<tr>
<td>APEXIR_NEXT_WEEK</td>
<td>Next Week</td>
</tr>
<tr>
<td>APEXIR_NEXT_X_DAYS</td>
<td>Next %0 Days</td>
</tr>
<tr>
<td>APEXIR_NEXT_X_HOURS</td>
<td>Next %0 Hours</td>
</tr>
<tr>
<td>APEXIR_NEXT_X_YEARS</td>
<td>Next %0 Years</td>
</tr>
<tr>
<td>APEXIR_NEXT_YEAR</td>
<td>Next Year</td>
</tr>
<tr>
<td>APEXIR_NO</td>
<td>No</td>
</tr>
<tr>
<td>APEXIR_NO_AGGREGATION_DEFINED</td>
<td>No aggregation defined.</td>
</tr>
<tr>
<td>APEXIR_NO_COLUMN_INFO</td>
<td>No column information available.</td>
</tr>
<tr>
<td>APEXIR_NO_COMPUTATION_DEFINED</td>
<td>No computation defined.</td>
</tr>
<tr>
<td>APEXIR_NO_END_DATE</td>
<td>- No End Date -</td>
</tr>
<tr>
<td>APEXIR_NONE</td>
<td>- None -</td>
</tr>
<tr>
<td>APEXIR_NOT_VALID_EMAIL</td>
<td>Not a valid email address.</td>
</tr>
<tr>
<td>APEXIR_NULL_SORTING</td>
<td>Null Sorting</td>
</tr>
<tr>
<td>APEXIR_NULLS_ALWAYS_FIRST</td>
<td>Nulls Always First</td>
</tr>
<tr>
<td>APEXIR_NULLS_ALWAYS_LAST</td>
<td>Nulls Always Last</td>
</tr>
<tr>
<td>APEXIR_NUMERIC_FLASHBACK_TIME</td>
<td>Flashback time must be numeric.</td>
</tr>
<tr>
<td>APEXIR_NUMERIC_SEQUENCE</td>
<td>Sequence must be numeric.</td>
</tr>
<tr>
<td>APEXIR_OPERATOR</td>
<td>Operator</td>
</tr>
<tr>
<td>APEXIR_ORANGE</td>
<td>orange</td>
</tr>
<tr>
<td>APEXIR_OTHER</td>
<td>Other</td>
</tr>
<tr>
<td>APEXIR_PERCENT_OF_TOTAL_COUNT_X</td>
<td>Percent of Total Count %0 (%)</td>
</tr>
</tbody>
</table>
### Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_PERCENT_OF_TOTAL_SUM_X</td>
<td>Percent of Total Sum %0 (%)</td>
</tr>
<tr>
<td>APEXIR_PERCENT_TOTAL_COUNT</td>
<td>Percent of Total Count</td>
</tr>
<tr>
<td>APEXIR_PERCENT_TOTAL_SUM</td>
<td>Percent of Total Sum</td>
</tr>
<tr>
<td>APEXIR_PIE</td>
<td>Pie</td>
</tr>
<tr>
<td>APEXIR_PIVOT</td>
<td>Add Pivot Column</td>
</tr>
<tr>
<td>APEXIR_PIVOT_AGG_NOT_NULL</td>
<td>Aggregate must be specified.</td>
</tr>
<tr>
<td>APEXIR_PIVOT_AGG_NOT_ON_ROW_COL</td>
<td>You cannot aggregate on a column selected to as row column.</td>
</tr>
<tr>
<td>APEXIR_PIVOT_COLUMN_N</td>
<td>Pivot Column %0</td>
</tr>
<tr>
<td>APEXIR_PIVOT_COLUMN_NOT_NULL</td>
<td>Pivot column must be specified.</td>
</tr>
<tr>
<td>APEXIR_PIVOT_COLUMNS</td>
<td>Pivot Columns</td>
</tr>
<tr>
<td>APEXIR_PIVOT_MAX_ROW_CNT</td>
<td>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.</td>
</tr>
<tr>
<td>APEXIR_PIVOT_SORT</td>
<td>Pivot Sort</td>
</tr>
<tr>
<td>APEXIR_PREVIOUS</td>
<td>&lt;</td>
</tr>
<tr>
<td>APEXIR_PRIMARY</td>
<td>Primary</td>
</tr>
<tr>
<td>APEXIR_PRIMARY_REPORT</td>
<td>Primary Report</td>
</tr>
<tr>
<td>APEXIR_PRIVATE</td>
<td>Private</td>
</tr>
<tr>
<td>APEXIR_PUBLIC</td>
<td>Public</td>
</tr>
<tr>
<td>APEXIR_RED</td>
<td>red</td>
</tr>
<tr>
<td>APEXIR_REMOVE</td>
<td>Remove</td>
</tr>
<tr>
<td>APEXIR_REMOVE_AGGREGATE</td>
<td>Remove Aggregate</td>
</tr>
<tr>
<td>APEXIR_REMOVE_ALL</td>
<td>Remove All</td>
</tr>
<tr>
<td>APEXIR_REMOVE_CHART</td>
<td>Remove Chart</td>
</tr>
<tr>
<td>APEXIR_REMOVE_CONTROL_BREAK</td>
<td>Remove Control Break</td>
</tr>
<tr>
<td>APEXIR_REMOVE_FILTER</td>
<td>Remove Filter</td>
</tr>
<tr>
<td>APEXIR_REMOVE_FLASHBACK</td>
<td>Remove Flashback</td>
</tr>
<tr>
<td>APEXIR_REMOVE_GROUP_BY</td>
<td>Remove Group By</td>
</tr>
<tr>
<td>APEXIR_REMOVE_HIGHLIGHT</td>
<td>Remove Highlight</td>
</tr>
<tr>
<td>APEXIR_REMOVE_PIVOT</td>
<td>Remove Pivot</td>
</tr>
<tr>
<td>APEXIR_RENAME_DEFAULT_REPORT</td>
<td>Rename Default Report</td>
</tr>
<tr>
<td>APEXIR_RENAME_REPORT</td>
<td>Rename Report</td>
</tr>
<tr>
<td>APEXIR_REPORT</td>
<td>Report</td>
</tr>
<tr>
<td>APEXIR_REPORT_ALIAS_DOES_NOT_EXIST</td>
<td>Saved Interactive report with alias %0 does not exist.</td>
</tr>
<tr>
<td>APEXIR_REPORT_DISPLAY_COLUMN_LIMIT_REACHED</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_REPORT_DOES_NOT_EXIST</td>
<td>Report does not exist.</td>
</tr>
<tr>
<td>APEXIR_REPORT_ID_DOES_NOT_EXIST</td>
<td>Saved Interactive Report ID %0 does not exist.</td>
</tr>
<tr>
<td>APEXIR_REPORT_SETTINGS</td>
<td>Report Settings</td>
</tr>
<tr>
<td>APEXIR_REPORT_VIEW</td>
<td>&lt; Report View</td>
</tr>
<tr>
<td>APEXIR_REPORTS</td>
<td>Reports</td>
</tr>
<tr>
<td>APEXIR_RESET</td>
<td>Reset</td>
</tr>
<tr>
<td>APEXIR_RESET_CONFIRM</td>
<td>Restore report to the default settings.</td>
</tr>
<tr>
<td>APEXIR_ROW</td>
<td>Row</td>
</tr>
<tr>
<td>APEXIR_ROW_COL_DIFF_FROM_PIVOT_COL</td>
<td>Row column must be different from the pivot column.</td>
</tr>
<tr>
<td>APEXIR_ROW_COLUMN_N</td>
<td>Row Column %0</td>
</tr>
<tr>
<td>APEXIR_ROW_COLUMN_NOT_NULL</td>
<td>Row column must be specified.</td>
</tr>
<tr>
<td>APEXIR_ROW_COLUMNS</td>
<td>Row Columns</td>
</tr>
<tr>
<td>APEXIR_ROW_FILTER</td>
<td>Row Filter</td>
</tr>
<tr>
<td>APEXIR_ROW_OF</td>
<td>Row %0 of %1</td>
</tr>
<tr>
<td>APEXIR_ROW_ORDER</td>
<td>Row Order</td>
</tr>
<tr>
<td>APEXIR_ROW_TEXT_CONTAINS</td>
<td>Row text contains</td>
</tr>
<tr>
<td>APEXIR_ROWS</td>
<td>Rows</td>
</tr>
<tr>
<td>APEXIR_ROWS_PER_PAGE</td>
<td>Rows Per Page</td>
</tr>
<tr>
<td>APEXIR_RPT_DISP_COL_EXCEED</td>
<td>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.</td>
</tr>
<tr>
<td>APEXIR_SAVE</td>
<td>Save</td>
</tr>
<tr>
<td>APEXIR_SAVE_AS_DEFAULT</td>
<td>Save as Default</td>
</tr>
<tr>
<td>APEXIR_SAVE_DEFAULT_CONFIRM</td>
<td>The current report settings are used as the default for all users.</td>
</tr>
<tr>
<td>APEXIR_SAVE_DEFAULT_REPORT</td>
<td>Save Default Report</td>
</tr>
<tr>
<td>APEXIR_SAVE_REPORT</td>
<td>Save Report</td>
</tr>
<tr>
<td>APEXIR_SAVE_REPORT_DEFAULT</td>
<td>Save Report *</td>
</tr>
<tr>
<td>APEXIR_SAVED_REPORT</td>
<td>Saved Report</td>
</tr>
<tr>
<td>APEXIR_SAVED_REPORT_MSG</td>
<td>Saved Report = &quot;%0&quot;</td>
</tr>
<tr>
<td>APEXIR_SEARCH</td>
<td>Search</td>
</tr>
<tr>
<td>APEXIR_SEARCH_BAR</td>
<td>Search Bar</td>
</tr>
<tr>
<td>APEXIR_SEARCH_COLUMN</td>
<td>Search: %0</td>
</tr>
<tr>
<td>APEXIR_SEARCH_REPORT</td>
<td>Search Report</td>
</tr>
<tr>
<td>APEXIR_SELECT_CATEGORY</td>
<td>- Select Category -</td>
</tr>
<tr>
<td>APEXIR_SELECT_COLUMN</td>
<td>- Select Column -</td>
</tr>
<tr>
<td>APEXIR_SELECT_COLUMNS</td>
<td>Select Columns</td>
</tr>
<tr>
<td>APEXIR_SELECT_COLUMNS_FOOTER</td>
<td>Computed columns are prefixed with **.</td>
</tr>
</tbody>
</table>
### Table 21–3 (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_SELECT_FUNCTION</td>
<td>- Select Function -</td>
</tr>
<tr>
<td>APEXIR_SELECT_GROUP_BY_COLUMN</td>
<td>- Select Group By Column -</td>
</tr>
<tr>
<td>APEXIR_SELECT_PIVOT_COLUMN</td>
<td>- Select Pivot Column -</td>
</tr>
<tr>
<td>APEXIR_SELECT_ROW</td>
<td>Select Row</td>
</tr>
<tr>
<td>APEXIR_SELECT_ROW_COLUMN</td>
<td>- Select Row Column -</td>
</tr>
<tr>
<td>APEXIR_SELECT_SORT_COLUMN</td>
<td>- Select Sort Column -</td>
</tr>
<tr>
<td>APEXIR_SELECT_VALUE</td>
<td>Select Value</td>
</tr>
<tr>
<td>APEXIR_SELECTED_COLUMNS</td>
<td>Selected Columns</td>
</tr>
<tr>
<td>APEXIR_SEND</td>
<td>Send</td>
</tr>
<tr>
<td>APEXIR_SEQUENCE</td>
<td>Sequence</td>
</tr>
<tr>
<td>APEXIR_SORT</td>
<td>Sort</td>
</tr>
<tr>
<td>APEXIR_SORT_ASCENDING</td>
<td>Sort Ascending</td>
</tr>
<tr>
<td>APEXIR_SORT_COLUMN</td>
<td>Sort Column</td>
</tr>
<tr>
<td>APEXIR_SORT_DESCENDING</td>
<td>Sort Descending</td>
</tr>
<tr>
<td>APEXIR_SORT_ORDER</td>
<td>Sort Order</td>
</tr>
<tr>
<td>APEXIR_SPACE_AS_IN_ONE_EMPTY_STRING</td>
<td>space</td>
</tr>
<tr>
<td>APEXIR_STATUS</td>
<td>Status</td>
</tr>
<tr>
<td>APEXIR_SUBSCRIPTION</td>
<td>Subscription</td>
</tr>
<tr>
<td>APEXIR_SUBSCRIPTION_ENDING</td>
<td>Ending</td>
</tr>
<tr>
<td>APEXIR_SUBSCRIPTION_STARTING_FROM</td>
<td>Starting From</td>
</tr>
<tr>
<td>APEXIR_SUBSCRIPTION_STARTING_FROM</td>
<td>Starting From</td>
</tr>
<tr>
<td>APEXIR_SUM_X</td>
<td>Sum %0</td>
</tr>
<tr>
<td>APEXIR_TABLE_SUMMARY</td>
<td>Region = %0, Report = %1, View = %2, Displayed Rows Start = %3, Displayed Rows End = %4, Total Rows = %5</td>
</tr>
<tr>
<td>APEXIR_TEXT_COLOR</td>
<td>Text Color</td>
</tr>
<tr>
<td>APEXIR_TIME_DAYS</td>
<td>days</td>
</tr>
<tr>
<td>APEXIR_TIME_HOURS</td>
<td>hours</td>
</tr>
<tr>
<td>APEXIR_TIME_MINS</td>
<td>minutes</td>
</tr>
<tr>
<td>APEXIR_TIME_MONTHS</td>
<td>months</td>
</tr>
<tr>
<td>APEXIR_TIME_WEEKS</td>
<td>weeks</td>
</tr>
<tr>
<td>APEXIR_TIME_YEARS</td>
<td>years</td>
</tr>
<tr>
<td>APEXIR_TOGGLE</td>
<td>Toggle</td>
</tr>
<tr>
<td>APEXIR_TOP</td>
<td>Top</td>
</tr>
<tr>
<td>APEXIR_UNAUTHORIZED</td>
<td>Unauthorized</td>
</tr>
<tr>
<td>APEXIR_UNGROUPED_COLUMN</td>
<td>Ungrouped Column</td>
</tr>
<tr>
<td>APEXIR_UNIQUE_COLUMN_HEADING</td>
<td>Column Heading must be unique.</td>
</tr>
<tr>
<td>APEXIR_UNIQUE_HIGHLIGHT_NAME</td>
<td>Highlight Name must be unique.</td>
</tr>
</tbody>
</table>
### Table 21–3  (Cont.) Interactive Report Messages Requiring Translation

<table>
<thead>
<tr>
<th>Message Name</th>
<th>English Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEXIR_UNSUPPORTED_DATA_TYPE</td>
<td>unsupported data type</td>
</tr>
<tr>
<td>APEXIR_UP</td>
<td>Up</td>
</tr>
<tr>
<td>APEXIR_VALID_COLOR</td>
<td>Please enter a valid color.</td>
</tr>
<tr>
<td>APEXIR_VALID_FORMAT_MASK</td>
<td>Please enter a valid format mask.</td>
</tr>
<tr>
<td>APEXIR_VALUE</td>
<td>Value</td>
</tr>
<tr>
<td>APEXIR_VALUE_AXIS_TITLE</td>
<td>Axis Title for Value</td>
</tr>
<tr>
<td>APEXIR_VALUE_REQUIRED</td>
<td>Value Required</td>
</tr>
<tr>
<td>APEXIR_VCOLUMN</td>
<td>Vertical Column</td>
</tr>
<tr>
<td>APEXIR_VIEW_CHART</td>
<td>View Chart</td>
</tr>
<tr>
<td>APEXIR_VIEW_DETAIL</td>
<td>View Detail</td>
</tr>
<tr>
<td>APEXIR_VIEW_GROUP_BY</td>
<td>View Group By</td>
</tr>
<tr>
<td>APEXIR_VIEW_ICONS</td>
<td>View Icons</td>
</tr>
<tr>
<td>APEXIR_VIEW_PIVOT</td>
<td>View Pivot</td>
</tr>
<tr>
<td>APEXIR_VIEW_REPORT</td>
<td>View Report</td>
</tr>
<tr>
<td>APEXIR_WEEK</td>
<td>Week</td>
</tr>
<tr>
<td>APEXIR_WEEKLY</td>
<td>Weekly</td>
</tr>
<tr>
<td>APEXIR_WORKING_REPORT</td>
<td>Working Report</td>
</tr>
<tr>
<td>APEXIR_X_DAYS</td>
<td>%0 days</td>
</tr>
<tr>
<td>APEXIR_X_HOURS</td>
<td>%0 hours</td>
</tr>
<tr>
<td>APEXIR_X_MINS</td>
<td>%0 minutes</td>
</tr>
<tr>
<td>APEXIR_X_MONTHS</td>
<td>%0 months</td>
</tr>
<tr>
<td>APEXIR_X WEEKS</td>
<td>%0 weeks</td>
</tr>
<tr>
<td>APEXIR_X_YEARS</td>
<td>%0 years</td>
</tr>
<tr>
<td>APEXIR_YEAR</td>
<td>Year</td>
</tr>
<tr>
<td>APEXIR_YELLOW</td>
<td>yellow</td>
</tr>
<tr>
<td>APEXIR_YES</td>
<td>Yes</td>
</tr>
<tr>
<td>IR_AS_DEFAULT_REPORT_SETTING</td>
<td>As Default Report Settings</td>
</tr>
<tr>
<td>IR_AS_NAMEDREPORT</td>
<td>As Named Report</td>
</tr>
<tr>
<td>IR_STAR</td>
<td>Only displayed for developers</td>
</tr>
<tr>
<td>LAYOUT.T_CONDITION_EXPR2</td>
<td>Expression 2</td>
</tr>
<tr>
<td>OUT_OF_RANGE</td>
<td>Invalid set of rows requested, the source data of the report has been modified.</td>
</tr>
<tr>
<td>REPORT</td>
<td>Report</td>
</tr>
<tr>
<td>REPORTING_PERIOD</td>
<td>Reporting Period</td>
</tr>
<tr>
<td>RESET</td>
<td>Reset Pagination</td>
</tr>
<tr>
<td>SAVED_REPORTS.PRIMARY.DEFAULT</td>
<td>Primary Default</td>
</tr>
<tr>
<td>WWV_RENDER_REPORT3.X_Y_OF_Z_2</td>
<td>%0 - %1 of %2</td>
</tr>
</tbody>
</table>
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.

See Also: APEX_LANG in Oracle Application Express API Reference

Defining a Dynamic Translation

You define dynamic translations on the Dynamic Translations page. A dynamic translation consists of a translate-from language string, a language code, and a translate-to string.

To define a dynamic translation:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the Application Builder icon.
   b. Select an application.
   c. Click Shared Components.
   d. Under Globalization, click Translate Application.

2. On the Translate Application page under Translation Utilities, select Dynamic Translations.

3. On the Dynamic Translations page, click Create and specify the following:
   a. Language - Select a target language.
   b. Translate From Text - Enter the source text to be translated.
   c. Translate To Text - Enter the translated text.

4. Click Create.

Understanding Supported Globalization Codes

If you are building a multilingual application, it is important to understand how globalization codes affect the way in which your application runs. These codes are set automatically based on the application-level Globalization attributes you select.

See Also: "Specifying the Primary Language for an Application" on page 21-4

Oracle Application Express Globalization Codes

Table 21–4 describes the globalization codes in Oracle Application Express. NLS_LANGUAGE and NLS_TERRITORY determine the default presentation of numbers, dates, and currencies.
<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>NLS_LANGUAGE</th>
<th>NLS_TERRITORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afrikaans</td>
<td>af</td>
<td>ENGLISH</td>
<td>SOUTH AFRICA</td>
</tr>
<tr>
<td>Arabic</td>
<td>ar</td>
<td>ARABIC</td>
<td>UNITED ARAB EMIRATES</td>
</tr>
<tr>
<td>Arabic (Algeria)</td>
<td>ar-dz</td>
<td>ARABIC</td>
<td>ALGERIA</td>
</tr>
<tr>
<td>Arabic (Bahrain)</td>
<td>ar-bh</td>
<td>ARABIC</td>
<td>BAHRAIN</td>
</tr>
<tr>
<td>Arabic (Egypt)</td>
<td>ar-eg</td>
<td>EGYPTIAN</td>
<td>EGYPT</td>
</tr>
<tr>
<td>Arabic (Iraq)</td>
<td>ar-iq</td>
<td>ARABIC</td>
<td>IRAQ</td>
</tr>
<tr>
<td>Arabic (Jordan)</td>
<td>ar-jo</td>
<td>ARABIC</td>
<td>JORDAN</td>
</tr>
<tr>
<td>Arabic (Kuwait)</td>
<td>ar-kw</td>
<td>ARABIC</td>
<td>KUWAIT</td>
</tr>
<tr>
<td>Arabic (Lebanon)</td>
<td>ar-lb</td>
<td>ARABIC</td>
<td>LEBANNON</td>
</tr>
<tr>
<td>Arabic (Libya)</td>
<td>ar-ly</td>
<td>ARABIC</td>
<td>LIBYA</td>
</tr>
<tr>
<td>Arabic (Morocco)</td>
<td>ar-ma</td>
<td>ARABIC</td>
<td>MOROCCO</td>
</tr>
<tr>
<td>Arabic (Oman)</td>
<td>ar-om</td>
<td>ARABIC</td>
<td>OMAN</td>
</tr>
<tr>
<td>Arabic (Qatar)</td>
<td>ar-qa</td>
<td>ARABIC</td>
<td>QATAR</td>
</tr>
<tr>
<td>Arabic (Saudi Arabia)</td>
<td>ar-sa</td>
<td>ARABIC</td>
<td>SAUDI ARABIA</td>
</tr>
<tr>
<td>Arabic (Syria)</td>
<td>ar-sy</td>
<td>ARABIC</td>
<td>SYRIA</td>
</tr>
<tr>
<td>Arabic (Tunisia)</td>
<td>ar-tn</td>
<td>ARABIC</td>
<td>TUNISIA</td>
</tr>
<tr>
<td>Arabic (U.A.E.)</td>
<td>ar-ae</td>
<td>ARABIC</td>
<td>UNITED ARAB EMIRATES</td>
</tr>
<tr>
<td>Arabic (YEMEN)</td>
<td>ar-ye</td>
<td>ARABIC</td>
<td>YEMEN</td>
</tr>
<tr>
<td>Assamese</td>
<td>as</td>
<td>ASSAMESE</td>
<td>INDIA</td>
</tr>
<tr>
<td>Basque</td>
<td>eu</td>
<td>FRENCH</td>
<td>FRANCE</td>
</tr>
<tr>
<td>Belarusian</td>
<td>be</td>
<td>RUSSIAN</td>
<td>RUSSIA</td>
</tr>
<tr>
<td>Bengali</td>
<td>bn</td>
<td>BANGLA</td>
<td>BANGLADESH</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>bg</td>
<td>BULGARIAN</td>
<td>BULGARIA</td>
</tr>
<tr>
<td>Catalaan</td>
<td>ca</td>
<td>CATALAN</td>
<td>CATALONIA</td>
</tr>
<tr>
<td>Chinese</td>
<td>zh</td>
<td>SIMPLIFIED CHINESE</td>
<td>CHINA</td>
</tr>
<tr>
<td>Chinese (China)</td>
<td>zh-cn</td>
<td>SIMPLIFIED CHINESE</td>
<td>CHINA</td>
</tr>
<tr>
<td>Chinese (Hong Kong SAR)</td>
<td>zh-hk</td>
<td>TRADITIONAL CHINESE</td>
<td>HONG KONG</td>
</tr>
<tr>
<td>Chinese (Macau SAR)</td>
<td>zh-mo</td>
<td>TRADITIONAL CHINESE</td>
<td>HONG KONG</td>
</tr>
<tr>
<td>Chinese (Singapore)</td>
<td>zh-sg</td>
<td>SIMPLIFIED CHINESE</td>
<td>SINGAPORE</td>
</tr>
<tr>
<td>Chinese (Taiwan)</td>
<td>zh-tw</td>
<td>TRADITIONAL CHINESE</td>
<td>TAIWAN</td>
</tr>
<tr>
<td>Croatian</td>
<td>hr</td>
<td>CROATIAN</td>
<td>CROATIA</td>
</tr>
<tr>
<td>Czech</td>
<td>cs</td>
<td>CZECH</td>
<td>CZECH REPUBLIC</td>
</tr>
<tr>
<td>Danish</td>
<td>da</td>
<td>DANISH</td>
<td>DENMARK</td>
</tr>
<tr>
<td>Dutch (Belgium)</td>
<td>nl-be</td>
<td>DUTCH</td>
<td>BELGIUM</td>
</tr>
<tr>
<td>Dutch (Netherlands)</td>
<td>nl</td>
<td>DUTCH</td>
<td>THE NETHERLANDS</td>
</tr>
<tr>
<td>English</td>
<td>en</td>
<td>AMERICAN</td>
<td>AMERICA</td>
</tr>
</tbody>
</table>
### Table 21–4 (Cont.) Oracle Application Express Globalization Codes

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>NLS_LANGUAGE</th>
<th>NLS_TERRITORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>English (Australia)</td>
<td>en-au</td>
<td>ENGLISH</td>
<td>AUSTRALIA</td>
</tr>
<tr>
<td>English (Belize)</td>
<td>en-bz</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>English (Canada)</td>
<td>en-ca</td>
<td>ENGLISH</td>
<td>CANADA</td>
</tr>
<tr>
<td>English (Ireland)</td>
<td>en-ie</td>
<td>ENGLISH</td>
<td>IRELAND</td>
</tr>
<tr>
<td>English (Jamaica)</td>
<td>en-jm</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>English (New Zealand)</td>
<td>en-nz</td>
<td>ENGLISH</td>
<td>NEW ZEALAND</td>
</tr>
<tr>
<td>English (Philippines)</td>
<td>en-ph</td>
<td>ENGLISH</td>
<td>PHILIPPINES</td>
</tr>
<tr>
<td>English (South Africa)</td>
<td>en-za</td>
<td>ENGLISH</td>
<td>SOUTH AFRICA</td>
</tr>
<tr>
<td>English (Trinidad)</td>
<td>en-tt</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>English (United Kingdom)</td>
<td>en-gb</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>English (United States)</td>
<td>en-us</td>
<td>ENGLISH</td>
<td>AMERICAN</td>
</tr>
<tr>
<td>English (Zimbabwe)</td>
<td>en-zw</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Estonian</td>
<td>et</td>
<td>ENGLISH</td>
<td>ESTONIAN</td>
</tr>
<tr>
<td>Faeroese</td>
<td>fo</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Farsi</td>
<td>fa</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Finnish</td>
<td>fi</td>
<td>ENGLISH</td>
<td>FINNISH</td>
</tr>
<tr>
<td>French (Belgium)</td>
<td>fr-be</td>
<td>ENGLISH</td>
<td>BELGIUM</td>
</tr>
<tr>
<td>French (Canada)</td>
<td>fr-ca</td>
<td>ENGLISH</td>
<td>CANADA</td>
</tr>
<tr>
<td>French (France)</td>
<td>fr</td>
<td>ENGLISH</td>
<td>FRANCE</td>
</tr>
<tr>
<td>French (Luxembourg)</td>
<td>fr-lu</td>
<td>ENGLISH</td>
<td>LUXEMBOURG</td>
</tr>
<tr>
<td>French (Monaco)</td>
<td>fr-mc</td>
<td>ENGLISH</td>
<td>FRANCE</td>
</tr>
<tr>
<td>French (Switzerland)</td>
<td>fr-ch</td>
<td>ENGLISH</td>
<td>SWITZERLAND</td>
</tr>
<tr>
<td>FYRO Macedonian</td>
<td>mk</td>
<td>ENGLISH</td>
<td>FYR MACEDONIA</td>
</tr>
<tr>
<td>Gaelic</td>
<td>gd</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Galician</td>
<td>gl</td>
<td>ENGLISH</td>
<td>SPAIN</td>
</tr>
<tr>
<td>German (Austria)</td>
<td>de-at</td>
<td>GERMAN</td>
<td>AUSTRIA</td>
</tr>
<tr>
<td>German (Germany)</td>
<td>de</td>
<td>GERMAN</td>
<td>GERMANY</td>
</tr>
<tr>
<td>German (Liechtenstein)</td>
<td>de-li</td>
<td>GERMAN</td>
<td>GERMANY</td>
</tr>
<tr>
<td>German (Luxembourg)</td>
<td>de-lu</td>
<td>GERMAN</td>
<td>LUXEMBOURG</td>
</tr>
<tr>
<td>German (Switzerland)</td>
<td>de-ch</td>
<td>GERMAN</td>
<td>SWITZERLAND</td>
</tr>
<tr>
<td>Greek</td>
<td>el</td>
<td>ENGLISH</td>
<td>GREECE</td>
</tr>
<tr>
<td>Gujarati</td>
<td>gu</td>
<td>GUJARATI</td>
<td>INDIA</td>
</tr>
<tr>
<td>Hebrew</td>
<td>he</td>
<td>HEBREW</td>
<td>ISRAEL</td>
</tr>
<tr>
<td>Hindi</td>
<td>hi</td>
<td>HINDI</td>
<td>INDIA</td>
</tr>
<tr>
<td>Hungarian</td>
<td>hu</td>
<td>HUNGARIAN</td>
<td>HUNGARY</td>
</tr>
<tr>
<td>Icelandic</td>
<td>is</td>
<td>ICELANDIC</td>
<td>ICELAND</td>
</tr>
</tbody>
</table>
Table 21–4 *(Cont.)* Oracle Application Express Globalization Codes

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>NLS_LANGUAGE</th>
<th>NLS_TERRITORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indonesian</td>
<td>id</td>
<td>INDONESIAN</td>
<td>INDONESIA</td>
</tr>
<tr>
<td>Italian (Italy)</td>
<td>it</td>
<td>ITALIAN</td>
<td>ITALY</td>
</tr>
<tr>
<td>Italian (Switzerland)</td>
<td>it-ch</td>
<td>ITALIAN</td>
<td>SWITZERLAND</td>
</tr>
<tr>
<td>Japanese</td>
<td>ja</td>
<td>JAPANESE</td>
<td>JAPAN</td>
</tr>
<tr>
<td>Kannada</td>
<td>kn</td>
<td>KANNADA</td>
<td>INDIA</td>
</tr>
<tr>
<td>Kazakh</td>
<td>kk</td>
<td>CYRILLIC KAZAKH</td>
<td>KAZAKHSTAN</td>
</tr>
<tr>
<td>Konkani</td>
<td>kok</td>
<td>KOREAN</td>
<td>KOREA</td>
</tr>
<tr>
<td>Korean</td>
<td>ko</td>
<td>KOREAN</td>
<td>KOREA</td>
</tr>
<tr>
<td>Kyrgyz</td>
<td>kz</td>
<td>RUSSIAN</td>
<td>RUSSIA</td>
</tr>
<tr>
<td>Latvian</td>
<td>lv</td>
<td>LATVIAN</td>
<td>LATVIA</td>
</tr>
<tr>
<td>Lithuanian</td>
<td>lt</td>
<td>LITHUANIAN</td>
<td>LITHUANIANA</td>
</tr>
<tr>
<td>Malay (Malaysia)</td>
<td>ms</td>
<td>MALAY</td>
<td>MALAYSIA</td>
</tr>
<tr>
<td>Malayalam</td>
<td>ml</td>
<td>MALAYALAM</td>
<td>INDIA</td>
</tr>
<tr>
<td>Maltese</td>
<td>mt</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Marathi</td>
<td>mr</td>
<td>ENGLISH</td>
<td>INDIA</td>
</tr>
<tr>
<td>Nepali (India)</td>
<td>ne</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Norwegian (Bokmal)</td>
<td>nb-no</td>
<td>NORWEGIAN</td>
<td>NORWAY</td>
</tr>
<tr>
<td>Norwegian (Bokmal)</td>
<td>no</td>
<td>NORWEGIAN</td>
<td>NORWAY</td>
</tr>
<tr>
<td>Norwegian (Nynorsk)</td>
<td>nn-no</td>
<td>NORWEGIAN</td>
<td>NORWAY</td>
</tr>
<tr>
<td>Oriya</td>
<td>or</td>
<td>ORIYA</td>
<td>INDIA</td>
</tr>
<tr>
<td>Polish</td>
<td>pl</td>
<td>POLISH</td>
<td>POLAND</td>
</tr>
<tr>
<td>Portuguese (Brazil)</td>
<td>pt-br</td>
<td>BRAZILIAN PORTUGUESE</td>
<td>BRAZIL</td>
</tr>
<tr>
<td>Portuguese (Portugal)</td>
<td>pt</td>
<td>PORTUGUESE</td>
<td>PORTUGAL</td>
</tr>
<tr>
<td>Punjabi</td>
<td>pa</td>
<td>PUNJABI</td>
<td>INDIA</td>
</tr>
<tr>
<td>Romanian</td>
<td>ro</td>
<td>ROMANIAN</td>
<td>ROMANIA</td>
</tr>
<tr>
<td>Russian</td>
<td>ru</td>
<td>RUSSIAN</td>
<td>RUSSIA</td>
</tr>
<tr>
<td>Russian (Moldova)</td>
<td>ru-md</td>
<td>RUSSIAN</td>
<td>RUSSIA</td>
</tr>
<tr>
<td>Serbia</td>
<td>sr</td>
<td>CYRILLIC SERBIAN</td>
<td>SERBIA AND MONTENEGRO</td>
</tr>
<tr>
<td>Slovak</td>
<td>sk</td>
<td>SLOVAK</td>
<td>SLOVAKIA</td>
</tr>
<tr>
<td>Slovenian</td>
<td>sl</td>
<td>SLOVENIAN</td>
<td>SLOVENIA</td>
</tr>
<tr>
<td>Spanish (Argentina)</td>
<td>es-ar</td>
<td>LATIN AMERICAN SPANISH</td>
<td>ARGENTINA</td>
</tr>
<tr>
<td>Spanish (Bolivia)</td>
<td>es-bo</td>
<td>LATIN AMERICAN SPANISH</td>
<td>ARGENTINA</td>
</tr>
<tr>
<td>Spanish (Chile)</td>
<td>es-cl</td>
<td>LATIN AMERICAN SPANISH</td>
<td>CHILE</td>
</tr>
<tr>
<td>Spanish (Columbia)</td>
<td>ec-co</td>
<td>LATIN AMERICAN SPANISH</td>
<td>COLUMBIA</td>
</tr>
<tr>
<td>Spanish (Costa Rica)</td>
<td>es-cr</td>
<td>LATIN AMERICAN SPANISH</td>
<td>COSTA RICA</td>
</tr>
<tr>
<td>Spanish (Dominican Republic)</td>
<td>es-do</td>
<td>LATIN AMERICAN SPANISH</td>
<td>PUERTO RICO</td>
</tr>
</tbody>
</table>

Managing Application Globalization 21-37
Table 21–4  (Cont.) Oracle Application Express Globalization Codes

<table>
<thead>
<tr>
<th>Language Name</th>
<th>Language Code</th>
<th>NLS_LANGUAGE</th>
<th>NLS_TERRITORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish (Ecuador)</td>
<td>es-ec</td>
<td>LATIN AMERICAN SPANISH</td>
<td>ECUDOR</td>
</tr>
<tr>
<td>Spanish (El Salvador)</td>
<td>es-sv</td>
<td>LATIN AMERICAN SPANISH</td>
<td>EL SALVADOR</td>
</tr>
<tr>
<td>Spanish (Guatemala)</td>
<td>es-gt</td>
<td>LATIN AMERICAN SPANISH</td>
<td>GUATEMALA</td>
</tr>
<tr>
<td>Spanish (Honduras)</td>
<td>es-hn</td>
<td>LATIN AMERICAN SPANISH</td>
<td>GUATEMALA</td>
</tr>
<tr>
<td>Spanish (Mexico)</td>
<td>es-mx</td>
<td>MEXICAN SPANISH</td>
<td>MEXICO</td>
</tr>
<tr>
<td>Spanish (Nicaragua)</td>
<td>es-ni</td>
<td>LATIN AMERICAN SPANISH</td>
<td>Nicaragua</td>
</tr>
<tr>
<td>Spanish (Panama)</td>
<td>es-pa</td>
<td>LATIN AMERICAN SPANISH</td>
<td>Panama</td>
</tr>
<tr>
<td>Spanish (Paraguay)</td>
<td>es-py</td>
<td>LATIN AMERICAN SPANISH</td>
<td>ARGENTINA</td>
</tr>
<tr>
<td>Spanish (Peru)</td>
<td>es-pe</td>
<td>LATIN AMERICAN SPANISH</td>
<td>PERU</td>
</tr>
<tr>
<td>Spanish (Puerto Rico)</td>
<td>es-pr</td>
<td>LATIN AMERICAN SPANISH</td>
<td>PUERTO RICO</td>
</tr>
<tr>
<td>Spanish (Traditional Sort)</td>
<td>es</td>
<td>LATIN AMERICAN SPANISH</td>
<td>SPAIN</td>
</tr>
<tr>
<td>Spanish (United States)</td>
<td>es-us</td>
<td>LATIN AMERICAN SPANISH</td>
<td>AMERICAN</td>
</tr>
<tr>
<td>Spanish (Uruguay)</td>
<td>es-uy</td>
<td>LATIN AMERICAN SPANISH</td>
<td>ARGENTINA</td>
</tr>
<tr>
<td>Spanish (Venezuela)</td>
<td>es-ve</td>
<td>LATIN AMERICAN SPANISH</td>
<td>VENEZUELA</td>
</tr>
<tr>
<td>Swedish</td>
<td>sv</td>
<td>SWEDISH</td>
<td>SWEDEN</td>
</tr>
<tr>
<td>Swedish</td>
<td>sv-fi</td>
<td>SWEDISH</td>
<td>FINLAND</td>
</tr>
<tr>
<td>Tamil</td>
<td>ta</td>
<td>TAMIL</td>
<td>INDIA</td>
</tr>
<tr>
<td>Telugu</td>
<td>te</td>
<td>TELUGU</td>
<td>INDIA</td>
</tr>
<tr>
<td>Thai</td>
<td>th</td>
<td>THAI</td>
<td>THAILAND</td>
</tr>
<tr>
<td>Turkish</td>
<td>tr</td>
<td>TURKISH</td>
<td>TURKEY</td>
</tr>
<tr>
<td>Ukrainian</td>
<td>uk</td>
<td>UKRAINIAN</td>
<td>UKRAINE</td>
</tr>
<tr>
<td>Urdu</td>
<td>ur</td>
<td>ENGLISH</td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Uzbek</td>
<td>uz</td>
<td>LATIN UZBEK</td>
<td>UZBEKISTAN</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>vi</td>
<td>VIETNAMESE</td>
<td>VIETNAM</td>
</tr>
</tbody>
</table>
22

Debugging an Application

This section describes approaches how to debug an application.

- Running Advisor to Check Application Integrity
- Reviewing Session State
- About Monitoring Application and Page Resource Use
- Enabling SQL Tracing and Using TKPROF
- Utilizing Debug Mode
- Removing Controls and Components to Isolate a Problem

**See Also:** "APEX_DEBUG" in *Oracle Application Express API Reference* and "Managing Application Performance" on page 23-1.

**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
- Running Advisor on an Entire Application
- Running Advisor on a Single Page

**About Oracle Application Express Advisor**

Before deploying your application, you can use the Oracle Application Express Advisor to perform various sanity checks on your application. The Advisor checks for errors, security issues, performance bottlenecks, quality assurance, and other best practices.

**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 *Application Builder* icon.
      The Application Builder home page appears.
b. Select the application.
   The Application home page appears.

2. Click Utilities.
3. Click Advisor.
   The Advisor page appears.
4. Under Checks to Perform:
   a. Expand Checks to Perform.
   b. Review the selected options. Select and deselect options as appropriate.
5. In Pages(s), enter a comma separated list of pages in the field provided. To check
   all pages, leave this option blank.
6. Click Perform Check at the top of the page.
   A Results page appears.

   Tip: Current Advisor settings are used next time the check is performed.

7. To alter the existing preferences:
   a. Under Filter Result, deselect the appropriate options.
   b. Click Apply Filter
8. To run Advisor again, click Perform Check.

Running Advisor on a Single Page

To run Advisor on a single page:
1. Navigate to the appropriate page.
   ■ In Page Designer, see “Viewing a Page in Page Designer” on page 8-15.
   ■ In Component View, see “Viewing a Page in Component View” on page 9-5.
2. Click the Utilities menu and select Advisor.
   Advisor appears.
3. Under Checks to Perform, review the selected options. Select and deselect options
   as appropriate.
4. Click Perform Check at the top of the page.
   A Results page appears.
5. To alter the existing preferences:
   a. Under Filter Result, deselect the appropriate options.
   b. Click Apply Filter
6. To re-run the Advisor and see if reported problems are fixed, click Perform Check.

Reviewing Session State

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" on page 9-25, "Viewing Session State" on page 2-15, "Managing Session State Values" on page 2-20, and "Managing Session State and User Preferences" in Oracle Application Express Administration Guide

**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.

**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.
5. 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

**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.
Utilizing Debug Mode

- Enabling and Disabling Debug Mode for an Application
- Running an Application in Debug Mode
- Viewing Debug Reports in Development Mode
- About Using f?p Syntax to Access Debug Mode
- Viewing Debug Reports

See Also:  "APEX_DEBUG" in Oracle Application Express API Reference

About Debug Mode

The Debug mode is a built-in mechanism used to track down unexpected application behavior. You enable and disable debug mode for the current page and session by clicking the Debug and No Debug links in the Runtime Developer Toolbar. To view Debug reports, click the View Debug link in the Runtime Developer Toolbar.

If an application runs in debug mode, Oracle Application Express writes information about how it processes each page request to a log table. You can control the logging level of detail, add debug calls in your own code, and analyze the written log information.

When debug is enabled, Oracle Application Express also uses its non-minified JavaScript files to better isolate where possible issues may be coming from with client-side code.

See Also:  "Runtime Developer Toolbar” on page 8-24 and "APEX_DEBUG” in Oracle Application Express API Reference

Enabling and Disabling Debug Mode for an Application

You can configure whether end users can run the application in debug mode by using the Debugging attribute on the Edit Application Definition page. A developer who is logged into workspace where the application resides can always run the application in debug mode.

To enable or disable debug mode:

1. On the Workspace home page, click the Application 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” on page 7-13
Utilizing Debug Mode

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 Application 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" on page 22-4

Viewing Debug Reports in Development Mode

To view debug reports in development mode:

1. On the Workspace home page, click the Application Builder icon.
2. Select the desired application.
3. Click Utilities.
   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" on page 22-4

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
Utilizing Debug Mode

See Also:  "About Using f?p Syntax to Link Pages" on page 2-17

Viewing Debug Reports from a Running Application

To view debug report from a running application:

1. Locate and run the application in Debug mode. See "Running an Application in Debug Mode" on page 22-5.

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:  "Enabling and Disabling Debug Mode for an Application" on page 22-4

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.
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 by selecting the Condition attribute Never.

To remove a control or component using conditional attributes:

1. Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
2. Select the name of the control or component you want to disable.
   The appropriate attributes page appears.
3. Scroll down to Condition Type and select Never from the Condition Type list.
4. Click Apply Changes and return to the page.
5. Try running the page again.
6. Continue to remove controls or components until the page runs correctly.

See Also:  "Enabling and Disabling Debug Mode for an Application" on page 22-4

See Also:  "Managing Pages in Page Designer" on page 8-1 and "Managing Pages in Component View" on page 9-1
Managing Application Performance

This section describes how to improve application performance.

- About Best Practices to Improve Performance
- Identifying Performance Issues
- Utilizing Logs and Reports
- About Utilizing Database Reporting
- About Database Parameters that Impact Performance
- About Limiting Resources
- About Uploading Static Files to Your Web Server
- Creating Custom Activity Reports Using APEX_ACTIVITY_LOG

See Also: "Debugging an Application" on page 22-1

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. See "Referencing Session State Using Bind Variable Syntax" on page 2-24.
- Include a #TIMING# substitution string in the region footer so that you can view the timing of each region. See "Understanding Substitution Strings" on page 2-25.

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
- Diagnosing Performance Issues
- About Debugging Problematic SQL Queries
- About Addressing Slow Queries
About Troubleshooting Slow Running Applications

The first step in troubleshooting a slow running application is to determine the location of the bottleneck. You can evaluate application performance by reviewing the time it takes for the database to return a specific page request or submission. If this time is shorter than the time it takes to render the page on the user's browser, then other components are likely causing the perception that the application is slow. As a best practice, always check all of the components involved including client machines, the network, the middle-tier, the database, and disk and storage devices. Regarding the database, check the System Global Area (SGA), CPU utilization, locks, and database parameters. It is also important to determine if anything else is running within the database such as legacy applications, data warehouse, batch programs, and database jobs.

The majority of the time, performance issues with Oracle Application Express applications are due to developer-authored SQL and PL/SQL. You should utilize standard database performance tuning techniques to address these issues. Performance issues are rarely related to a bug with Oracle Application Express or the Oracle database.

See Also: Oracle Database Performance Tuning Guide and Oracle Database SQL Tuning Guide

Diagnosing Performance Issues

To diagnose performance issues, follow these steps:

- Step 1: Identify the time window.
- Step 2: Isolate the largest consumers of database time.
- Step 3: Identify the Oracle Application Express workspace and application.
- Step 4: Correlate the offending SQL with a specific location within the application.

Once you have identified the slow running pages, you should run the application in debug mode to identify the specific components consuming the most time.

About Debugging Problematic SQL Queries

If your query does not seem to be running correctly, try running it in SQL*Plus, SQL Developer, or in SQL Commands. Any of these approaches will test your query outside the context of your application, making it easier to define the problem.

About Addressing Slow Queries

For optimized queries that still take more than a second to process, consider implementing a progress bar and preventing multiple page submissions. Users expect instant results and often resubmit a page if it does not return immediately. Unless managed properly, each page submission instigates a new session in the database and continues until completion. Multiple page submissions often degrade performance and may lead to database locks. This is especially true with interactive reports. As an alternative approach, try using Oracle Application Express collections to hold query results. By using collections, expensive and time consuming queries are only executed once for each user rather than each time they perform pagination or create a filter.

See Also: "Editing Page Attributes in Component View" on page 9-11, "Duplicate Submission" on page 9-19, and "About Using Collections" on page 25-6
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.

- How to Manage Log Files
- Viewing the Application Dashboard Report
- Workspace Activity Reports
- Instance Activity Reports

**Tip:** Another method of determining expensive elements within a page or process is to review the Application Express Debug mode. "APEX_DEBUG" in Oracle Application Express API Reference

**See Also:** "Utilizing Debug Mode" on page 22-3

How to Manage Log Files

An Instance administrator can delete activity log files and manage the interval at which log file information is retained. If you wish to keep the activity log files indefinitely, you must create a batch job to copy records from the activity log table to another table.

**See Also:** "Managing Logs and Files" in Oracle Application Express Administration Guide

Viewing the Application Dashboard Report

To view the Application Dashboard:

1. Navigate to the Workspace home page.
2. Click the Application 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.
Utilizing Logs and Reports

- **Application Components** contains links to reports of application-level controls and logic, including lists of values, tabs, lists, application items, application processes, and application computations.

- **Page Components** contains links to reports of page-level controls and logic, including items, buttons, processes, regions, branches, validations, and dynamic actions.

6. To view a specific report, click the number to the right of the component type or template.

**Workspace Activity Reports**

Workspace administrators are users who perform administrator tasks specific to a workspace. 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 as 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* and "About Application Express User Roles" on page 1-3

**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.
About Database Parameters that Impact Performance

- **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 Application Express User Roles" on page 1-3

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_ROLE then you can query across the entire instance, rather than just the workspace your schema user is associated with.

*See Also:* "Enabling SQL Tracing and Using TKPROF" on page 22-3, Oracle Database Performance Tuning Guide for information on AWR and ASH data, and Oracle Application Express SQL Workshop Guide

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_050000 schema. Accurate statistics allow the
cost-based optimizer to develop better query plans for accessing the application metadata.

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.


See Also: "Managing Resources with Oracle Database Resource Manager" in Oracle Database Administrator’s Guide

About Uploading Static Files to Your Web Server

To improve performance you should place static files on the Web server. As a best practice, you should also enable gzip compression on the Web server so that the file size is minimized when transmitting the static files to the browser. Furthermore, you should also enable file caching for browsers by sending expiration headers.

See Also: "Managing Static Application Files" on page 18-8 and "Managing Static Workspace Files" on page 18-12

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
- APEX_ACTIVITY_LOG
- APEX_ACTIVITY_LOG Sample Query

Enabling Logging for an Application

You enable logging for an application on the Edit Application Definition page.

To edit the application definition:
1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. Click the Edit Application Properties button to the right of the application name. The Edit Application Definition page appears.

4. Under Properties, locate Logging. Select Yes or No.

When set to Yes, every page view is logged, enabling an administrator to monitor user activity for each application. Disabling logging may be advisable for high volume applications.

**Tip:** You can only modify the Logging attribute if the Application Activity Logging attribute in Oracle Application Express Administration Services is set to Use Application Setting.

**See Also:** "Enabling Application Activity" in Logging in Oracle Application Express Administration Guide

**APEX_ACTIVITY_LOG**

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 23–1 describes the columns in the APEX_ACTIVITY_LOG view.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time_stamp</td>
<td>DATE</td>
<td>Date and time that activity was logged at the end of the page view.</td>
</tr>
<tr>
<td>component_type</td>
<td>VARCHAR2(255)</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>component_name</td>
<td>VARCHAR2(255)</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>component_attribute</td>
<td>VARCHAR2(4000)</td>
<td>Title of page.</td>
</tr>
<tr>
<td>information</td>
<td>VARCHAR2(4000)</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>elap</td>
<td>NUMBER</td>
<td>Elapsed time of page view in seconds.</td>
</tr>
<tr>
<td>num_rows</td>
<td>NUMBER</td>
<td>Number of rows processed on page.</td>
</tr>
<tr>
<td>userid</td>
<td>VARCHAR2(255)</td>
<td>User ID performing page view.</td>
</tr>
<tr>
<td>ip_address</td>
<td>VARCHAR2(4000)</td>
<td>IP address of client.</td>
</tr>
<tr>
<td>ir_report_id</td>
<td>NUMBER</td>
<td>Interactive report ID</td>
</tr>
<tr>
<td>ir_search</td>
<td>VARCHAR2</td>
<td>Interactive report search criteria entered by users.</td>
</tr>
<tr>
<td>user_agent</td>
<td>VARCHAR2(4000)</td>
<td>web browser user agent of client.</td>
</tr>
<tr>
<td>flow_id</td>
<td>NUMBER</td>
<td>Application ID.</td>
</tr>
<tr>
<td>step_id</td>
<td>NUMBER</td>
<td>Page number.</td>
</tr>
<tr>
<td>session_id</td>
<td>NUMBER</td>
<td>Oracle Application Express session identifier.</td>
</tr>
<tr>
<td>sqlerrm</td>
<td>VARCHAR2(4000)</td>
<td>SQL Error message.</td>
</tr>
<tr>
<td>sqlerrm_component_type</td>
<td>VARCHAR2(255)</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>sqlerrm_component_name</td>
<td>VARCHAR2(255)</td>
<td>Reserved for future use.</td>
</tr>
</tbody>
</table>
To conserve space in the activity log, only the first log entry of each unique session contains the IP address and web browser user agent.

**APEX_ACTIVITY_LOG Sample Query**

The following example demonstrates how to create a report that displays the total number of page views and the average page view time in the past 24 hours for application 9529, and grouped by userid:

```sql
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.
This section describes the steps involved in deploying a database application built within Application Builder.

- About Suggested Environments
- System Development Life Cycle Methodologies to Consider
- Understanding the Deployment Process
- How to Move an Application to Another Instance
- How to Create a Custom Packaged Application
- Exporting an Application and Application Components
- Importing Export Files
- Installing Export Files
- About Publishing the Database Application URL
- About Publishing the Websheet Application URL
- Using Build Options to Control Configuration

See Also: "Extending Application Capabilities" on page 19-1 and "Managing Application Express Users" in Oracle Application Express Administration Guide


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 Application 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" on page 1-2

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
- About the Advantages of Creating Prototypes
- Waterfall
- Spiral
- Rapid Application Development

About Iterative Vs Planned Development

When developing applications using Application Builder, you must find a balance between two dramatically different development methodologies:

- Iterative, rapid application development
- Planned, linear style development

Iterative, rapid application development offers so much flexibility that you run the risk of never completing your project. In contrast, Planned, linear style development can yield applications that do not meet the needs of end users even if they meet the stated requirements on paper.

About the Advantages of Creating Prototypes

The Oracle Application Express development environment enables developers to take a more iterative approach to development. Unlike many other development environments, creating prototypes is easy. With Oracle Application Express, developers can:

- Use built-in wizards to quickly design an application user interface.
- Make prototypes available to users and gather feedback.
- Implement changes in real time, creating new prototypes instantly.

Methodologies that work well with Oracle Application Express include Spiral and Rapid Application Development (RAD).

See Also:  "Spiral" on page 24-3 and "Rapid Application Development" on page 24-3

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
Understanding the Deployment Process

3. Design
4. Development
5. Integration and Testing
6. Installation and Acceptance
7. Maintenance

This methodology is referred to as a waterfall because the output from one stage is the input for the next stage.

A primary problem with this approach is that it is assumed that all requirements can be established in advance. Unfortunately, requirements often change and evolve during the development process.

Spiral

A Spiral methodology is actually a series of short waterfall cycles. Each waterfall cycle yields new requirements and enables the development team to create a robust series of prototypes. One advantage of this approach is that it accommodates changing requirements. Disadvantages include complex project management and the risk development goes on indefinitely.

Rapid Application Development

A Rapid Application Development (RAD) methodology has a heavy emphasis on creating a prototype that closely resembles the final product. The prototype is an essential part of the requirements phase. Advantages of this model include the ability to accommodate changing requirements, rapid development cycles, and progress can be easily measured. The major disadvantage of this model is that the emphasis on prototyping can result in scope creep. As a result, developers can lose sight of their initial goals in the attempt to create the perfect application.

Understanding the Deployment Process

This section discusses different approaches to the deployment process.

- About Moving an Application to Another Instance
- Deployment Options to Consider
- About Whether to Copy a Workspace
- About Whether to Copy a Database
- About the Application ID

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. See "How to Move an Application to Another Instance" on page 24-5.
2. Move the supporting objects. Review the Database Object Dependencies report to determine what objects to move. See "Using the Database Object Dependencies Report" on page 7-64 and "How to Create a Custom Packaged Application" on
Deployment Options to Consider

When you develop an application in Application Builder, you create the application within a specific workspace. Each workspace has a unique ID and name. A common scenario is 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. See "Using the Database Object Dependencies Report" on page 7-64.

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.

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.

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.

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.
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. See "Using the Database Object Dependencies Report" on page 7-64.

2. Package the application definition with its supporting objects to create a custom packaged application. See "How to Create a Custom Packaged Application" on page 24-6.

3. Export the custom packaged application. See "Exporting an Application and Application Components" on page 24-12.

4. Import the exported files into the target Oracle Application Express instance. See "Importing Export Files" on page 24-23.

You can import an application into your workspace regardless of the workspace in which it was developed. See “Deployment Options to Consider” on page 24-4.

Tip: You can also move the application definition and all supporting objects manually. See "Exporting an Application and Application Components" on page 24-12.

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.

See Also: "Using the Database Object Dependencies Report" on page 7-64 and "How to Create a Custom Packaged Application" on page 24-6

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.
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
- Creating a Custom Packaged Application
- Adding an Access Control List to a Custom Packaged Application
- Installing Supporting Objects
- Deleting Supporting Objects Scripts, Messages, and Installation Options
- Upgrading a Custom Packaged Application
- Deinstalling Supporting Objects
- Viewing an Install Summary

See Also: "How to Move an Application to Another Instance" on page 24-5 and "Exporting an Application and Application Components" on page 24-12

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. See “Installing Supporting Objects” on page 24-9.
- Enables users and developers with a convenient method for removing the application definition, supporting files, and all database objects. See “Deinstalling Supporting Objects” on page 24-11.
- Provides users and developers with an easy way to upgrade a previously released custom packaged application. See "Upgrading a Custom Packaged Application"
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
- About the Supporting Objects Page

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 Application Builder icon.
2. Select the application.
   The Application home page appears.
3. Click the Supporting Objects icon.
   The Supporting Objects page appears.

About the Supporting Objects Page

The top of the Supporting Objects page displays the application name and indicates current selections for the following: Check for Objects, Substitutions, Installation Scripts, Verify System Privileges, Required Free KB, Prompt for License, Validations, Deinstallation Script, Prompt for License, and Include in Export. To learn more about these options, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

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.
The rest of the page is divided into three categories.

**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. See "Substitutions" on page 7-19.

  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. See "Using Build Options to Control Configuration" on page 24-31 and "Exporting Build Options or Build Option Status" on page 24-32.

  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 <b>, <i>, <u>, <p>, <br>, <hr>, <ul>, <ol>, <li>, and <pre>.

  When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack. See "Understanding Cross-Site Scripting Protection" on page 20-16.

**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 "Upgrading a Custom Packaged Application" on page 24-10.

- **Upgrade Message.** Enables you to define messages that display when the user upgrades the application. Supported HTML tags include <b>, <i>, <u>, <p>, <br>, <hr>, <ul>, <ol>, <li>, and <pre>.

  When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack. See "Understanding Cross-Site Scripting Protection" on page 20-16.

**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 `<b>`, `<i>`, `<u>`, `<p>`, `<br>`, `<hr>`, `<ul>`, `<ol>`, `<li>`, and `<pre>`.

  When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack. See "Understanding Cross-Site Scripting Protection" on page 20-16.

### 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:


2. Navigate to the Supporting Objects page:
   a. On the Workspace home page, click the **Application 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, 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**.

### 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 **Application 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, click **Preview Installation Script**.
   
   The Preview Scripts page appears listing summary information, prerequisites, and the actual scripts to be run.

6. To exit the Preview Scripts page and continue, click **Close**.

7. From Install Supporting Objects, click **Yes** and click **Next**.

8. Follow the on-screen instructions.

### Deleting Supporting Objects Scripts, Messages, and Installation Options

You can delete the metadata that defines supporting object scripts, messages, and installation options associated with a custom 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 **Application Builder** icon.
2. Select an application.
3. Click **Supporting Objects**.
   
   The Supporting Objects page appears.
4. On the Tasks list on the right side of the page, click **Remove Supporting Object Installation**.
5. Follow the on-screen instructions.

### Upgrading a Custom Packaged Application

You can define scripts to upgrade a previously published application on the Upgrade page.

- **Defining an Upgrade Script**
- **Upgrading a Custom Packaged Application**

#### Defining an Upgrade Script

You can use the Upgrade page to define scripts to upgrade database objects, images, and seed data when upgrading an existing application.

To create an upgrade script:

1. On the Workspace home page, click the **Application 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.

**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). See "Importing an Application, Page or Component Export" on page 24-23.

2. On the Workspace home page, click the Application 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.

**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.

**See Also:** "Creating a Custom Packaged Application" on page 24-7.

To deinstall a custom packaged application:

1. On the Workspace home page, click the Application 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.

**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 **Application 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.

**Exporting an Application and Application Components**

This section describes how to export and application and related files to another Oracle Application Express instance.

- **About the Export Process**
- **About The Export Page**
- **Exporting an Application**
- **Exporting Application Components**
- **Exporting a Workspace**
- **Exporting a Page in an Application**
- **Exporting a Websheet**
- **About Exporting Static Files**
- **Exporting Themes**
- **Exporting Plug-ins**
- **Exporting User Interface Defaults**
- **Exporting Team Development Feedback**
- **Exporting Script Files from the Script Repository**
Tip: You can also export workspaces, applications, and related files from a command-line using the APEXExport program located in the Utilities folder in the downloaded installation zip file. To learn more, see the Readme.txt file in the Utilities folder.

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. However, an application exported from Oracle 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, it is better to 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 then you can utilize Build Options to include or exclude various application components.

See Also: Exporting an Application, Exporting Application Components, and Using Build Options to Control Configuration

About The Export Page

You export an application definition and all associated files using the following tabs at the top of the Export page: Export, Workspace, Application, Websheet, Themes, Plug-ins, Defaults, and Feedback.
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 “How to Create a Custom Packaged Application” on page 24-6.

**See Also:** "How to Move an Application to Another Instance" on page 24-5

**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).

**See Also:** "Exporting Application Components" on page 24-16 and "Installing Exported Applications into a Runtime Environment" in Oracle Application Express Administration Guide

To export an application:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **Application Builder** icon.
b. On the Application 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 include:
      - **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 the application to the file repository instead of producing a file to download.
   b. **Owner Override** - Optionally select another owner. By default, the application is owned by the owner identified in the application attribute owner.
   c. **Build Status Override** - Select one of the following:
      - **Run Application Only** - Developers can only run an application.
      - **Run and Build Application** - 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**, you cannot set the argument `p_trace` to **Yes**. See "Using Build Options to Control Configuration" on page 24-31. Also, the only way to change this setting after you import the application, is to log in to Oracle Application Express Administration Services. See "Changing Application Build Status Set During Deployment" in Oracle Application Express Administration Guide.

d. **Debugging** - Select one of the following:
   - **Yes** - Exports the application with debugging enabled.
   - **No** - Exports the application with debugging disabled.

e. **As of** - Enables you to go back in time and export a previous version of your application. For example, you can use this option to export a version of your application that contain an object that is currently deleted. Specify the number of minutes in the field provided.

   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, select **Yes** or **No** to include or exclude the following:
   ■ **Export Supporting Object Definitions**
Exporting an Application and Application Components

- Export Public Interactive Reports
- Export Private Interactive Reports
- Export Interactive Report Subscriptions
- Export Developer Comments
- Export Translations
- Export with Original IDs

To learn more about an attribute, see field-level Help. See “Viewing Field-Level Help” on page 1-21.

6. Click Export Application.

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: "Enabling SQL Tracing and Using TKPROF" on page 22-3 and "Importing an Application, Page or Component Export" on page 24-23

Exporting Application Components

You can export shared components or components of a page on the Component Export page. You can use this wizard to:

- Export shared components or page components to another application or workspace.
- Back up a component before editing it.
- Create an export to function 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 Application 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. See “Working with Shared Components” on page 18-1.
   - Components by Page lists components of the selected page. Navigate to a specific page by making a selection from the Page. Click Check All to select all components.
   - Build Option Status displays available build options. Use this page to turn build options on and off. See “Exporting Build Options or Build Option Status” on page 24-32.

   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 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 to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

      This utility uses the DBMS_FLASHBACK package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter UNDO_RETENTION (the default is three hours). However, this only influences the size of the undo tablespace. While two databases 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: "Exporting an Application" on page 24-14, "Exporting a Page in an Application" on page 24-18, "Importing an Application, Page or Component Export" on page 24-23, and "Exporting Build Options or Build Option Status" on page 24-32

**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 and ASCII text SQL script of users, any defined user groups, SQL Workshop artifacts like saved SQL, history, and SQL Scripts, and optionally Team Development data.

To export a workspace:

1. Navigate to the Export page:
   a. On the Workspace home page, click the Application Builder icon.
   b. On the Application Builder home page, click Workspace Utilities.
   c. Click Export.

2. On the Export page, click the Workspace tab.

3. Under Export Workspace:
   ■ Include Team Development - Select Yes to include Team Development data such as Features, Milestones, To Do, Bugs, Feedback, Links and News. Only
Exporting an Application and Application Components

select Yes if the target instance for this workspace is not a runtime only instance. Otherwise, select No.

■ 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.

4. Click Export.

See Also: "Managing Application Express Users" in Oracle Application Express Administration Guide

Exporting a Page in an Application

This section describes how to export a page in an application. When exporting a page, remember that exported pages can only be imported successfully if they have the same application ID and workspace ID.

To export a page in an application:

1. Navigate to the appropriate page:
   a. Page Designer:
      - Navigate to the appropriate page. See "Viewing a Page in Page Designer" on page 8-15.
      - Click the Utilities menu and select Export.
   b. Component View:
      - Navigate to the appropriate page. See "Viewing a Page in Component View" on page 9-5.
      - Click the Utilities menu and select Export.

The Export Page Wizard appears.

2. From Page, select the page to export.

3. 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.

4. 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.

5. Click Export Page.
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 Application Builder icon.
   b. On the Application Builder home page, click Workspace Utilities.
   c. Click Export.
2. On the Export page, click the Websheet tab.
3. On the Websheets page:
   a. Websheet Application - Select the Websheet to export.
   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. Export Preferences - Select the appropriate options.
   d. Click Export.

See Also: "Creating Websheet Applications" on page 6-1 and "Importing a Websheet" on page 24-25

About Exporting Static Files

You can export static application files and static workspace files by creating a ZIP file. To learn more, see "Downloading All Static Application Files in a Zip" on page 18-11 and "Downloading All Static Workspace Files in a Zip" on page 18-14.

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 Application Builder icon.
   b. On the Application Builder home page, click Workspace Utilities.
   c. Click Export.
2. Click the Themes tab.
3. On the Themes page:
   a. Application - Select an application and click Set 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 rows delimited by line feeds.
   - DOS - The resulting file contains rows delimited by carriage returns and line feeds.

d. Click Export Theme.

See Also: "Importing Themes" on page 24-26 and "Using Themes" on page 13-1

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 Application Builder icon.
   b. On the Application 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 and click Set 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" on page 24-25

Exporting User Interface Defaults

This section describes how to export User Interface Defaults.

- About Exporting User Interface Defaults
- Exporting User Interface Defaults from the Export Page
- Exporting User Interface Defaults from the User Interface Defaults Page

About Exporting User Interface Defaults

Exporting User Interface Defaults is useful when you plan to develop on a target system. User Interface Defaults are divided into two categories, the Table Dictionary, and the Attribute Dictionary.

- The Table Dictionary consists is specific to tables and columns within a selected schema. These defaults are used over those defined in the Attribute Dictionary.
- The Attribute Dictionary contains a set of attributes about a column that is used in creating forms and reports. The definitions are matched by column name and a particular definition can be shared among several columns by using synonyms.

When you export User Interface Defaults, all User Interface Defaults for the selected dictionary and schema are exported to a single SQL Command script. When prompted, save this file to your hard drive. The file contains an API call to create table hints by making calls to the application PL/SQL API. You can use this file to import User Interface Defaults to another database and Oracle Application Express instance.

See Also: "Importing User Interface Defaults" on page 24-26 and "Managing User Interface Defaults" in Oracle Application Express SQL Workshop Guide

Exporting User Interface Defaults from the Export Page
To export User Interface Defaults from the Export page:
1. Navigate to the Export page:
   a. On the Workspace home page, click the Application Builder icon.
   b. On the Application 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 rows delimited by line feeds.
      - DOS - The resulting file contains rows delimited by carriage returns and line feeds.
   c. Click Export.
5. On the Attribute Dictionary page:
   a. File Format - Select how rows in the export file are formatted. Options include:
      - UNIX - The resulting file contains rows delimited by line feeds.
      - DOS - The resulting file contains rows delimited by carriage returns and line feeds.
   b. Click Export.

Exporting User Interface Defaults from the User Interface Defaults Page
To export User Interface Defaults from the User Interface Defaults page:
1. On the Workspace home page, click the SQL Workshop icon.
2. Click Utilities.
3. Click User Interface Defaults.
   The User Interface Defaults page appears.
4. Select the dictionary
■ Manage Table Dictionary - Select this button to export specific tables and columns within a selected schema.

■ Manage Attribute Dictionary - Select this button to export a set of column attributes used to create forms and reports.

5. Under Tasks, click Export.

6. For File format, select an option:
   - UNIX - The resulting file contains rows delimited by line feeds.
   - DOS - The resulting file contains rows delimited by carriage returns and line feeds.
   - Database - Exports to the file repository instead of producing a file to download.

7. Click Export.

Exporting Team Development Feedback

Use the Export Feedback utility to synchronize feedback from a production or test system to the development system. The development system is identified by using the workspace setting Feedback Synchronization Source Identifier which is set on the Edit Workspace Information page. If exporting from a development system, the export process has an additional attribute called Deployment System where you specify to which deployment system, production or test, you want to synchronize the updated feedback to.

To export Team Development Feedback:

1. Navigate to the Export page:
   a. On the Workspace home page, click the Application Builder icon.
   b. On the Application 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" on page 24-27

Exporting Script Files from the Script Repository

You can transfer selected scripts from your current Script Repository to a Script Repository in a different Workspace by using the Export and Import tasks.

To export script files:

1. On the Workspace home page, click the SQL Workshop icon.
2. Click SQL Scripts.
3. On the Tasks list, click Export.
4. Select the appropriate script files and click Add to Export.
5. Review the file name and click Export All.
   Select the Remove check box to remove the script.

See Also: "Using SQL Scripts" in Oracle Application Express SQL Workshop Guide

Importing Export Files

Once you export an application and any related files, you must import them into the target Oracle Application Express instance 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. See "How to Create a Custom Packaged Application" on page 24-6.

- Importing an Application, Page or Component Export
- About Importing Application Groups
- Importing a Websheet
- Importing Plug-ins
- Importing Themes
- Importing User Interface Defaults
- Importing Team Development Feedback

See Also: "How to Move an Application to Another Instance" on page 24-5

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 Application Builder icon.
   b. On the Application 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. See "Changing Application Build Status Set During Deployment" in *Oracle Application Express Administration Guide*.

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:** "How to Create a Custom Packaged Application" on page 24-6

**About Importing Application Groups**

Application groups are exported with an application, application exports include an group ID, name, and comments. When importing an application, the application import follows these rules:

- On import, an application uses an existing group if the ID matches.
- If no application group exists with the same ID on the target system, then application import looks for another group ID with the same name. If a name match is found, the application is imported with the ID corresponding to this application group name.
- If no ID or name matches existing application groups on the target system then a new application group is created. The new application group uses the application group ID if that ID is not used in that Oracle Application Express instance. If the group ID is being used, then a new ID is generated.
Importing a Websheet

See Also: "Creating Websheet Applications" on page 6-1

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 Application Builder icon.
   b. On the Application 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.

Importing Plug-ins

To import an exported plug-in file:
1. Navigate to the Import page:
   a. On the Workspace home page, click the Application Builder icon.
   b. On the Application 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.
Importing Export Files

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 Application Builder icon.
   b. On the Application 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" on page 13-1 and "Exporting Themes" on page 24-19

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 Application Builder icon.
   b. On the Application 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* and "Exporting User Interface Defaults" on page 24-20

### Importing Team Development Feedback

To import Team Development Feedback:

1. Navigate to the Import page:
   a. On the Workspace home page, click the **Application Builder** icon.
   b. On the Application Builder home page, click **Import**.

2. Select **Feedback**.

3. For File Character Set, verify that File Character Set is correct and click **Next**.
   Once you import a file, you have the option to install it.

4. To install an imported file, click **Next**.

5. Click **Install Feedback**.

### Installing Export Files

This section describes how to install export files.

- **About Installing Export Files**
- **Accessing the Export Repository**
- **Installing an Application Export from the Export Repository**
- **Installing Other Files from the Export Repository**
- **Deleting Files from the Export Repository**
- **Installing Export Files from SQL*Plus**
- **Using SQL*Plus to Install Export Files**

### About Installing Export Files

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 Application 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 "How to Create a Custom Packaged Application" on page 24-6.
See Also: "Managing Exported Files" and "Installing Exported Applications into a Runtime Environment" in Oracle Application Express Administration Guide

Accessing the Export Repository

When you import an application and any related files into a target Oracle Application Express instance, the files are stored in the Export Repository. You can use the Export Repository to manage archived application exports, imports, and other files.

To access the Export Repository:

1. On the Workspace home page, click the Application Builder icon.
2. Select an application.
3. On the Application home page, click Utilities.
4. Click Export Repository.

The Export Repository appears.

Installing an Application Export from the Export Repository

After you import an application export into an Oracle Application Express instance, you must install it before it can become active or available in Application Builder.

To install an application export from 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.
   c. Install As Application - Select one of the following:
      - Auto Assign New Application ID
      - Reuse Application ID from Export File
      - Change Application ID

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. See "Changing Application Build Status Set During Deployment" in Oracle Application Express Administration Guide.

See Also: "Managing Exported Files" and "Installing Exported Applications into a Runtime Environment" in Oracle Application Express Administration Guide
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: “How to Create a Custom Packaged Application” on page 24-6

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 Application Builder.

To install files stored in the Export Repository:

2. Select the file to be installed and click Install in the Action column.
3. Follow the on-screen instructions.

Deleting Files from the Export Repository

You can delete a file from the Export Repository.

To delete a file from the Export Repository:

2. Select the file to be deleted and click Delete Checked.

Installing Export Files from SQL*Plus

You can also install export files from SQL*Plus. The export file must originate from the same user database account as the one into which you are installing.

The following restrictions apply for export files:

- If the export file is a database application, then the application ID is overwritten. Therefore, the target workspace must own the ID of the application being installed.
- If the export file is a Websheet application, then the target workspace must have Websheet Database objects set-up.
- If the export file is a Websheet application, then the export file must be installed using the Websheet schema in the target workspace.
- If the export file is a Websheet application, then the application ID is overwritten.

Using SQL*Plus to Install Export Files

To install Oracle Application Express export files from SQL*Plus:
1. Log in to SQL*Plus.
2. Run the export file.
   For example, if your export file is names f144.sql by default, you would type `@f144` at the command prompt.

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:

```
```

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:

```
```

**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" on page 7-14

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:

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:


**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.

**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
- Managing Build Options
- Selecting a Build Option
- Exporting Build Options or Build Option Status
- Viewing the Build Option Utilization Report

**See Also:** “Changing Build Status for Multiple Applications” in Oracle Application Express Administration Guide and discussions of the GET_BUILD_OPTION_STATUS and SET_BUILD_OPTION_STATUS procedures in "APEX_UTIL" in Oracle Application Express API Reference

**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 **Application 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.

### 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 **Application 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**.

### 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:**  
["Editing Page Attributes in Component View" on page 9-11](#)

### 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.

**See Also:**  
["How to Create a Custom Packaged Application" on page 24-6](#) and ["Exporting Application Components" on page 24-16](#)

To export build options or build option status:
1. Navigate to the Component Export page:
   a. On the Workspace home page, click the Application 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.

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 Application 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.
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.

- Creating Applications with Data Loading Capability
- About Using Collections
- Implementing Web Services
- Accessing Data with Database Links
- About DML Locking

Creating Applications with Data Loading Capability

Applications with data loading capability enable end users to dynamically import data into a table within any schema to which the user has 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. This section explains how to create a Data Load Wizard.

- About Creating a Data Load Wizard
- Creating a Data Load Wizard
- Re-creating a Data Load Pages
- Editing the Data Load Definition

See Also: "Data Loading Wizard Examples" in Oracle Application Express End User Guide to learn about using the Data Load wizard.

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.

About Creating a Data Load Wizard

An application developer creates a Data Load Wizard by creating a Data Load page with the Create Page Wizard. During the creation process, developers can specify
upload table and its unique columns, table lookups, and data transformation rules. The Create Page Wizard creates a Data Load table that defines the Data Load Wizard.

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.

- 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.

- 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 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).

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.

See Also: "Creating a Data Load Wizard" on page 25-2 and "Editing the Data Load Definition" on page 25-5

Creating a Data Load Wizard

To create a Data Load Wizard:

1. On the Workspace home page, click the Application Builder icon.
2. Select the application.
3. Click Create Page and click Next.
4. For Create a Page:
   a. User Interface - Select a user interface for the page.
   b. Select a page type - Select Data Loading.
   c. Click Next.

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.
   h. Unique Column 3 - 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 third unique key column during the data load process.
   i. Case Sensitive - Identify whether the selected unique key column is case sensitive. By default, this is set to No.
   j. Click Next.

6. For Add Transformation Rules (optional), specify the following then click Add.
   a. Sequence - Specify the sequence for the transformation rule. The sequence determines the order of execution.
   b. Rule Name - Enter a name for this transformation rule.
   c. Select Columns - Select the name of the column that requires transformation and move to the right.
   d. Type - Select the type of transformation rule you want to perform.
   e. Click Next.

7. Repeat the previous step for each transformation rule you want to add, then click Next.

8. For Table Lookups (optional):
   a. Column Name - Identify the column on which the table lookup definition is to be based.
   b. Owner - Select the owner of the lookup table.
   c. Table Name - Identify the table to be used for this table lookup definition.
d. Return Column - Select the name of the column returned by the table lookup. This value will be inserted into the load column specified, and is generally the key value of the parent in a foreign key relationship.

e. Upload Column - Select the name of the column end users will upload instead of the return column. This is the column that contains the display value from the lookup table.

f. Upload Column 2 - Select the name of the second column to be uploaded to uniquely identify the return column if necessary.

g. Upload Column 3 - Select the name of the third column to be uploaded to uniquely identify the return column if necessary.

h. Click Next.

9. Repeat the previous step for each Table Lookup you want to add and click Next. The Page Attributes page appears providing a short description of each page of the Data Load Wizard being created.

10. For Pages Attributes, you can optionally change the default page names, page numbers and region names for each page of the data load wizard and click Next.

11. For Navigation Menu, specify the type of navigation to include on this page and click Next. The navigation options depend upon the current application theme.

12. For Buttons and Branching, specify the branching for the buttons on the data load wizard pages and click Next.

13. Review the Confirm page and click Create.

Page Designer appears.

14. Click Save and Run Page to test the Data Load Wizard.

**Tip:** After creating Data Load Wizard pages, if you wish to make changes, Oracle recommends re-creating new pages without deleting the data loading definitions as described in the next section.

**See Also:** "Re-creating a Data Load Pages" on page 25-4 and "Managing Pages in Page Designer" on page 8-1

**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 Application Builder.
   b. Select an application.
   c. On the Application home page, click Shared Components.
      The Shared Components page appears.
   d. Under Data References, click Data Load Definitions.

2. Click the Data Load Definition you want to re-create.

The Data Load Table Details page appears.
3. From Tasks, click Re-create Data Load Pages.
   The Page Attributes page of the Create Data Load Wizard appears.

4. For each page, edit the appropriate attributes and click Next.
   To learn more about any attributes, see field-level Help.

5. Click Confirm.

**Editing the Data Load Definition**

A Data Load Definition is comprised of a data load table, table rules, and lookup tables used by the Data Load Wizard in your application. A data load table is an existing table in your schema that has been selected for use in the data loading process, to upload data.

To edit a Data Load Definition:

1. Navigate to the Data Load Definitions page:
   a. On the Workspace home page, click **Application Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
      The Shared Components page appears.
   d. Under Data References, click **Data Load Definitions**.

2. Click the Data Load Definition you want to edit.
   The Data Load Table Details page appears. To learn more about any attributes, see field-level Help.

3. For Data Load Definition:
   - Name - Name for the Data Load Definition.
   - Data Load Table - Edit the data load table name and add, remove or edit unique columns.
   - Unique column 1 - The column name used as the primary unique key column during the data load process.
   - Case Sensitive - Identify whether the selected unique key column 1 is case sensitive. By default, this is set to No.
   - Unique column 2 - If the unique key definition of the selected table is a compound key, consisting of 2 or more columns, this column name is used as the second unique key column during the data load process.
   - Case Sensitive - Identify whether the selected unique key column 2 is case sensitive. By default, this is set to No.
   - Unique column 3 - If the unique key definition of the selected table is a compound key, consisting of 2 or more columns, this column name is used as the third unique key column during the data load process.
   - Case Sensitive - Identify whether the selected unique key column 3 is case sensitive. By default, this is set to No.
   - Skip Validation - One step in data loading is to validate actions to be taken on records to be uploaded. Select Yes to skip validation.

4. Transformation Rules lists previously defined transformation rules.
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*

---

### Implementing Web Services

This section describes how to utilize Web services in Oracle Application Express applications.

- **About Web Services**
- **Creating Web Service References in Oracle Application Express**
- **About Working with SSL Enabled Web Services**
- **Creating Web Service References Based on a WSDL**
- **Creating Web Service References Manually**
- **Creating RESTful Web Service References**
- **Using the Web Service Reference Repository**
- **Creating an Input Form and Report on a Web Service**
- **Creating a Form on a Web Service**
- **How to Invoke a Web Service as a Process**
- **Editing a Web Service Process**
- **Viewing a Web Service Reference History**
- **Exposing a Report Region as a RESTful Web Service**

---

**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

Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment. In a typical Web services scenario, a business application sends a request to a service at a given URL by using the protocol over HTTP. The service receives the request, processes it, and returns a response. You can incorporate calls with external Web services in applications developed in Application Builder.
Web services are typically based on Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) architectures. 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. See "Enabling Network Services in Oracle Database 11g or Later" on page 20-10

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.

### 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
- Specifying an Application Proxy Server Address

**See Also:** Creating Web Service References Based on a WSDL, Creating Web Service References Manually, and Creating RESTful Web Service References

### Accessing the Web Service References Page

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

To access the Web Service References page:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **Application 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.
Specifying an Application Proxy Server Address
If your environment requires a proxy server to access the Internet, you must specify a proxy server address on the Application Attributes page before you can create a Web service reference.

To specify a proxy address for an application:
1. On the Workspace home page, click the Application Builder icon.
   The Application Builder home page appears.
2. Select an application.
   Application home page appears.
3. Click Edit Application Properties.
4. Under Properties, enter the proxy server in the Proxy Server field.
5. Click Apply Changes.

About Working with SSL Enabled Web Services
Secure Sockets Layer (SSL) is an industry standard protocol that uses RSA public key cryptography with symmetric key cryptography to provide authentication, encryption, and data integrity.

If the Web service that you need to interact with is SSL-enabled (that is, https displays in the URL to the Web service), you must create a wallet and configure Oracle Application Express to use the wallet. A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL.

See Also: "Configuring Wallet Information” in Oracle Application Express Administration Guide

Creating Web Service References Based on a WSDL
This section describes how to create Web service reference based on a WSDL.

- About Creating Web Service References Based on a WSDL
- Creating a Web Service Reference by Specifying a WSDL Document
- Testing a Web Service Reference Created from a WSDL

About Creating Web Service References Based on a WSDL
Before you create a Web service reference based on a WSDL, you must decide how to locate the WSDL. You can locate a WSDL in two ways:

- By searching a Universal Description, Discovery and Integration (UDDI) registry
- By entering the URL to the WSDL document

Tip: A UDDI registry is a directory where businesses register their Web services.

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

Creating a Web Service Reference by Specifying a WSDL Document
To create a Web service by specifying a URL to a specific WSDL document:
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. See "Creating a Form After Creating a Reference" on page 25-17.

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:
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.

Creating Web Service References Manually

This section describes how to create a Web service reference manually.

- About Creating Web Service References Manually
- Creating a Web Service Reference Manually
- Testing a Web Service Reference Created Manually

About Creating Web Service References Manually

When you create a Web service reference manually, you supply the necessary information to create a valid SOAP request, including:

- The URL used to post the SOAP request over HTTP(S)
- A Universal Resource Identifier (URI) identifying the SOAP HTTP request
- The SOAP envelope for the request, including any item substitutions
- Optionally the name of a collection to store the response from the Web service

Creating a Web Service Reference Manually

To create a Web service reference manually:


2. Click Create.


   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.
Implementing Web Services

---

**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**.

The Web service reference is added to the Web Service References Repository.

**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:


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.

**Creating RESTful Web Service References**

This section describes how to create a RESTful Web service reference.

- About Creating RESTful Web Service References
- Creating a RESTful Web Service Reference
- Testing a REST Web Service Reference

**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

**Creating a RESTful Web Service Reference**

To create a RESTful Web service reference:

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. See "Creating a Proxy Server for an Instance" in Oracle Application Express Administration Guide.
   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.
Implementing Web Services

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 CLOB column of a collection you specify.

Type - Select the output type.

Click Add Parameter.

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.

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:


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.

Using the Web Service Reference Repository

Web service references are stored in the Web Service Reference Repository.

To access the Web Service References Repository:


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.
2. Implementing Web Services

Managing Application Data

- 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. See "About the Actions Menu" on page 10-58.

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.

Creating an Input Form and Report on a Web Service

This section describes how to create an input form and report on a Web service.

- About the Create Form and Report on Web Service Wizard
- Creating a Form and Report After Creating a Reference
- Creating a Form and Report by Adding a New Page

About the Create Form and Report on Web Service Wizard

The Create Form and Report on Web Service Wizard creates an input form, a submit button, and a report for displaying results. You can execute this wizard directly after creating the Web service reference from a WSDL or a RESTful style Web service, or by adding a page.

Use this wizard when you expect a nonscalar result from the Web service. The Amazon Product API Web service is a good example. This Web service returns many results based on the search criteria entered in an input form.

Creating a Form and Report After Creating a Reference

To create a form and report after creating a 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. See "Managing the Application User Interface" on page 7-29.

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**.

### 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:


2. Create a new page. See "Managing Pages in a Database Application" on page 7-36.

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. See "Managing the Application User Interface" on page 7-29.
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**.

### Creating a Form on a Web Service

This section describes how to create a form on a Web service.

- **About the Create Form on Web Service Wizard**
- **Creating a Form After Creating a Reference**
- **Creating a Form by Adding a New Page**

#### About the Create Form on Web Service Wizard

The Create Form on Web Service Wizard creates a form and a submit button. You can execute this wizard after creating the Web service reference from a WSDL or on a RESTful style Web service, or by running the Create Page Wizard.

Use this wizard when you expect a scalar result from the Web service. A Web service that looks up a stock price is a good example because the input is a stock symbol and the output is the scalar value price.

#### Creating a Form After Creating a Reference

To create a form after creating a Web service reference:
Implementing Web Services


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. See "Managing the Application User Interface" on page 7-29.

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.
   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.
Creating a Form by Adding a New Page

If you have an existing Web service reference, you can create a form based on a Web service reference by adding a new page.

To create a form by adding a new page:


2. Create a new page. See "Managing Pages in a Database Application" on page 7-36.

   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. See "Managing the Application User Interface" on page 7-29.
   b. Select Form and click Next.
   c. Select Form on Web Service and click Next.

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.

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.
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.

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
- Displaying Web Service Results in a Report

Invoking a Web Service as a Process
To invoke a Web service as a process:

2. Create a page. See "Managing Pages in a Database Application" on page 7-36.
   In the Create Page Wizard:
   a. Select Blank Page.
   b. If prompted to use a navigation menu, select No.
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.

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.

Displaying Web Service Results in a Report
To create a report in which to display Web Service request results:

1. Navigate to the page. See "Viewing a Page in Component View" on page 9-5.
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.

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:


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.

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 Application Builder icon.
2. Select an application.
3. Click Shared Components.
   The Shared Components page appears.
5. Click History.

---

Note: The History button only appears on the Web Service Reference page if at least one reference exists.

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
- Enabling RESTful Access to a Report Region
- About Accessing a RESTful Enabled Report Region from a Web Service Client

See Also: "Using RESTful Services" in Oracle Application Express SQL Workshop Guide

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. To enable RESTful access to an Oracle Application Express instance, see "Enabling RESTful Access" in the Oracle Application Express Administration Guide.
- Step 3: The page that contains the report must not require authentication.
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 Application Builder.
   The Application Builder home page appears.

2. Select an application.

3. View the page in Component View that contains the report you want to enable.
   See "Viewing a Page in Component View" on page 9-5.

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.
   d. Click Apply Changes.

**Note:** The Enable RESTful Access List option only displays if RESTful access to this Oracle Application Express instance has been enabled, see "Enabling RESTful Access" in the Oracle Application Express Administration Guide.

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.
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
<?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>P1_JOB</urn:value>
        <urn:value>P1_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.

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td>N/A</td>
<td>Yes</td>
<td>The numeric ID or alias of the application that contains the RESTful enabled report.</td>
</tr>
<tr>
<td>page</td>
<td>N/A</td>
<td>Yes</td>
<td>The numeric ID or alias of the page that contains the RESTful enabled report.</td>
</tr>
<tr>
<td>reportid</td>
<td>N/A</td>
<td>Yes</td>
<td>The Static ID attribute of the RESTful enabled report.</td>
</tr>
<tr>
<td>parmvalues</td>
<td>null</td>
<td>No</td>
<td>Values for the report parameters can be sent in a comma separated list. For example: CLERK,10</td>
</tr>
<tr>
<td>lang</td>
<td>en</td>
<td>No</td>
<td>Sets the NLS environment variables before running the report based on the language passed in. For example: de</td>
</tr>
<tr>
<td>output</td>
<td>xml</td>
<td>Yes</td>
<td>This determines whether XML or JSON are returned to the client. Domain of possible values are: xml, json</td>
</tr>
</tbody>
</table>
Accessing Data with Database Links

This section describes how to access data by creating a database link.

- About Database Links
- Creating a Database Link
- Viewing an Existing Database Link

See Also: "Managing Database Objects with Object Browser" in Oracle Application Express SQL Workshop Guide and "Database Links" in Oracle Database Administrator’s Guide

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 @dbname to the table or view name where dbname is the Database Link Name you specify in the Create Database Object Wizard.

Note: By default, the CREATE DATABASE LINK system privilege is not granted to a provisioned workspace or database user. To use this feature, a DBA or administrator must grant this specific privilege to the database user in the user's workspace. See "Creating Database Links" in Oracle Database Administrator’s Guide.

Creating a Database Link

To create a database link:

1. On the Workspace home page, click SQL Workshop and then Object Browser.
   Object Browser home page appears.
2. Click Create.
3. Select Database Link and click Next.
4. Follow the on-screen instructions.
   Note that Database Link names must conform to Oracle naming conventions and cannot contain spaces, or start with a number or underscore.

Viewing an Existing Database Link

To view an existing a database link:

1. On the Workspace home page, click SQL Workshop and then Object Browser.
   Object Browser appears.
2. Select the object type Database Links at the top of the page.
About DML Locking

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
- APEX_DML_LOCK_WAIT_TIME
- FSP_DML_LOCK_ROW

About DML Locking

When you use automatic Data Manipulation Language (DML) in Oracle Application Express to update or delete rows of a table, a transaction is initiated to first lock the row, verify if it has changed since it was displayed on the page, and then finally issue the actual UPDATE or DELETE statement for the row.

In some environments where locking of rows is prevalent, you may want to control the DML operation and determine if the DML operation:

- waits indefinitely
- fails immediately
- waits for a specified period of time

APEX_DML_LOCK_WAIT_TIME

You can set the value of an application substitution string, an application item, or a page item to APEX_DML_LOCK_WAIT_TIME to control the DML operation. APEX_DML_LOCK_WAIT_TIME supports the following values:

- NULL (the default), results in the same behavior as previous versions of Oracle Application Express, that is, wait indefinitely.
- 0 fails immediately if the row is locked by another database session.
- > 0 and the row is locked, waits for the specified number of seconds.

When set in an application, the value for APEX_DML_LOCK_WAIT_TIME applies to all UPDATE and DELETE DML operations using Automatic DML in the entire application. To control a specific Automatic DML process, update the value of APEX_DML_LOCK_WAIT_TIME before the Automatic DML process and reset it after the Automatic DML process. Note that this does not affect updates and deletes using tabular forms.

FSP_DML_LOCK_ROW

You can also set the value of an application substitution string, an application item, or a page item to FSP_DML_LOCK_ROW to control the DML operation. FSP_DML_LOCK_ROW supports the following values:

- If the value is set to FALSE, then no SELECT FOR UPDATE is issued.
- If the value is anything other than FALSE, the default behavior of SELECT FOR UPDATE is performed when issuing an UPDATE or DELETE DML operation using Automatic DML.
About DML Locking
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. You can control how a page item works by editing the page item attributes. Each item type in Table A–1, "Available Item Types" describes key attributes that control item functionality. To learn more about all item attributes, see field-level Help.

See Also:  "Understanding Page-Level Items" on page 15-1, "Editing Page-Level Items in Page Designer" on page 15-12, and "Editing Page-Level Items in Component View" on page 15-23

Available Item Types

Table A–1, "Available Item Types" describes available item types in Application Builder and notes important attributes that control item behavior.
<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Box</td>
<td>Displayed using a list of values. A list of values is required for items displayed as check boxes. The value corresponding to a checked box is returned in a single colon-delimited string. The following example demonstrates how to create a single check box that returns YES. This example would display both a check box and a field label. SELECT NULL display_text, 'YES' return_value FROM DUAL; This example includes the additional text Click to select. SELECT 'Click to select' display_text, 'YES' return_value FROM DUAL;</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;APEX_UTIL&quot; in Oracle Application Express API Reference for information about breaking up returned values</td>
</tr>
<tr>
<td></td>
<td>Page Designer item attributes:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Validation, Value Required</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Settings, Number Columns</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> “Creating a Cascading List of Values” on page 15-10</td>
</tr>
<tr>
<td></td>
<td>Component View item attributes:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Settings, Value Required</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Settings, Number of Columns</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> “Creating a Cascading List of Values” on page 15-20</td>
</tr>
<tr>
<td>Color Picker</td>
<td>Renders as a text field with an icon. When the user clicks the icon, a popup window appears. When the user makes a selection from the palette, the HTML value for the color selected (for example, #000000 for black) is returned. <strong>Note:</strong> When creating a new item of this type, first select Popup List of Values and then Color Picker.</td>
</tr>
<tr>
<td></td>
<td>Page Designer item attributes:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Validation, Value Required</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td>Component View item attributes:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Settings, Value Required</strong> - 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.</td>
</tr>
</tbody>
</table>
Table A–1 (Cont.) Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Picker</td>
<td>Displays a text field with a Calendar icon next to it. When clicked, this icon displays a small calendar where the user can select a date and a time (optional).</td>
</tr>
<tr>
<td></td>
<td>Page Designer item attributes:</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Minimum Date - Enter the minimum date that can be input or selected.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Maximum Date - Enter the maximum date that can be input or selected.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Show - Select when the date picker pop-up calendar displays.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Navigation List for - Select whether select lists are provided within the calendar pop-up for the month and/or year.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Highlighted Date - Enter a default date value to be highlighted when the calendar pop-up displays. See Help for more details.</td>
</tr>
<tr>
<td></td>
<td>■ Appearance, Format Mask - Enter a date format mask to apply to this component.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>Component View item attributes:</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Minimum Date - Enter the minimum date that can be input or selected.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Maximum Date - Enter the maximum date that can be input or selected.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Show - Select when the date picker pop-up calendar displays.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ Settings, Navigation List - Select whether select lists are provided within the calendar pop-up for the month and/or year.</td>
</tr>
<tr>
<td>Date Picker (Classic)</td>
<td>Displays a text field with a Calendar icon next to it. When clicked, this icon displays a small calendar where the user can select a date and a time (optional).</td>
</tr>
</tbody>
</table>
|                 | If the format you need is not included in the Display As list, select Date Picker (use application format mask) or Date Picker (use item format mask). The latter uses the value from the Format Mask field in the Source section of the Edit Page Item page.
### Display Image

Enables the display of images stored in an image URL stored in another page item or in a BLOB column. To alter where the image is stored, edit the item and configure the Based on attribute. Options include:

- **Page Designer and Component View item attributes:**
  - **Settings, Based On** - Determines where the item is retrieved from. To learn more about an attribute, see field-level Help.
  - **Settings, Alternative Text Column** - Enter the uppercase page or application item name which holds the alternative text for this image. This text is read by screen readers when an end user navigates to this item.
  - **Settings, Filename Column** - Enter the uppercase page or application item name that holds the filename column value associated with the BLOB column.
  - **Settings, BLOB Last Updated Column** - If specified, the column is used for browser caching. It identifies if the image changed since the last browser request. If not entered, no caching is done. Note: This attribute is case sensitive and the column must be of type `DATE`.

### Display Only

Displays the item as display only text.

- **Page Designer and Component View item attributes:**
  - **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:
    - **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.
  - **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.

---

**Table A–1 (Cont.) Available Item Types**

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Image</td>
<td>Enables the display of images stored in an image URL stored in another page</td>
</tr>
<tr>
<td></td>
<td>item or in a BLOB column. To alter where the image is stored, edit the item</td>
</tr>
<tr>
<td></td>
<td>and configure the Based on attribute. Options include:</td>
</tr>
<tr>
<td></td>
<td>- Page Designer and Component View item attributes:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Settings, Based On</strong> - Determines where the item is retrieved from.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Settings, Alternative Text Column</strong> - Enter the uppercase page</td>
</tr>
<tr>
<td></td>
<td>or application item name which holds the alternative text for this</td>
</tr>
<tr>
<td></td>
<td>image. This text is read by screen readers when an end user navigates</td>
</tr>
<tr>
<td></td>
<td>to this item.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Settings, Filename Column</strong> - Enter the uppercase page or application</td>
</tr>
<tr>
<td></td>
<td>item name that holds the filename column value associated with the</td>
</tr>
<tr>
<td></td>
<td>BLOB column.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Settings, BLOB Last Updated Column</strong> - If specified, the column is</td>
</tr>
<tr>
<td></td>
<td>used for browser caching. It identifies if the image changed since the</td>
</tr>
<tr>
<td></td>
<td>last browser request. If not entered, no caching is done. Note: This</td>
</tr>
<tr>
<td></td>
<td>attribute is case sensitive and the column must be of type <code>DATE</code>.</td>
</tr>
<tr>
<td>Display Only</td>
<td>Displays the item as display only text.</td>
</tr>
<tr>
<td></td>
<td>- Page Designer and Component View item attributes:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Settings, Save Session State</strong> - Stores the current value in</td>
</tr>
<tr>
<td></td>
<td>session state when the page gets submitted.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Settings, Based On</strong> - Specifies what gets displayed in the field.</td>
</tr>
<tr>
<td></td>
<td>Options include:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Page Item Value</strong> - The current value of the page item displays.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Display Value of List of Values</strong> - The current value of the</td>
</tr>
<tr>
<td></td>
<td>page item maps to a different display value using a list of values.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Output of PL/SQL Code</strong> - The specified PL/SQL code takes</td>
</tr>
<tr>
<td></td>
<td>care of emitting the necessary output for the page item.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Settings, Show Line Breaks</strong> - Select Yes to convert line</td>
</tr>
<tr>
<td></td>
<td>breaks in the display value to an HTML line break and display them</td>
</tr>
<tr>
<td></td>
<td>in HTML output.</td>
</tr>
</tbody>
</table>
Available Item Types

Table A–1 (Cont.) Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Browse</td>
<td>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.</td>
</tr>
</tbody>
</table>

See Also: “About Securing File Uploads” on page 20-32

Page Designer attributes:
- Settings, Storage Type - Identify where the uploaded file is stored.
- Settings, MIME Type Column - Specify the name of the column where the MIME 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.
- 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.

Component View attributes:
- 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.
- Settings, MIME Type Column - Specify the name of the column where the MIME 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

Creates a hidden page item.

Page Designer and Component View attribute:
- Settings, Value is Protected - Select Yes to prevent hidden values from being manipulated when a page is posted.
Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Manager</td>
<td>Based on a list of values. This item enables you to manage a list of items by selecting and adding to a list. The list of values display as a popup. Page Designer attributes:</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Fetch</strong> - Select how data is displayed when the Popup LOV is opened.</td>
</tr>
<tr>
<td></td>
<td><strong>Validation, Value Required</strong> - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating a Cascading List of Values&quot; on page 15-10</td>
</tr>
<tr>
<td></td>
<td>Component View attributes:</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Value Required</strong> - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted.</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Fetch</strong> - Select how data is displayed when the Popup LOV is opened.</td>
</tr>
<tr>
<td></td>
<td><strong>See Also:</strong> &quot;Creating a Cascading List of Values&quot; on page 15-20</td>
</tr>
<tr>
<td>Number Field</td>
<td>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. Page Designer attributes:</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Minimum Value</strong> - Enter the minimum value permitted. The value can be a static value or a dynamic value using substitution syntax.</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Maximum Value</strong> - Enter the maximum value permitted. The value can be a static value or a dynamic value using substitution syntax.</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Number Alignment</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Appearance, Format Mask</strong> - Enter a number format mask to apply to this component. You can type in the format mask or pick from the list of predefined format masks. You can also use substitution syntax to reference a format mask from an application or page item.</td>
</tr>
<tr>
<td></td>
<td><strong>Validation, Value Required</strong> - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted.</td>
</tr>
<tr>
<td></td>
<td>Component View attributes:</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Value Required</strong> - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted.</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Format Mask</strong> - 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Minimum Value</strong> - Enter the minimum value permitted. The value can be a static value or a dynamic value using substitution syntax.</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Maximum Value</strong> - Enter the maximum value permitted. The value can be a static value or a dynamic value using substitution syntax.</td>
</tr>
<tr>
<td></td>
<td><strong>Settings, Number Alignment</strong> - 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.</td>
</tr>
</tbody>
</table>
### Table A–1 (Cont.) Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Renders as an HTML password form element.</td>
</tr>
</tbody>
</table>

**Page Designer attributes:**

- **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.

  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**.

- **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.

**Component View attributes:**

- **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.

  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**.

See Also: “About Session State and Security” on page 20-20.
Available Item Types

Table A–1  (Cont.) Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Popup List of Values (LOV)         | Renders as a text field with an icon. When the user clicks the icon, a popup window appears with a list of values represented as a series of links. When the user makes a selection from the list, the selected value is placed in the text field. You control popup lists of values through templates. You can only specify one popup list of values (LOV) template for each application. A popup LOV is a good choice for lists of values that are too large to return on a single page. Popup LOVs must be based on a query that selects two columns with different column aliases. For example:

```sql
SELECT ename name, empno id
FROM emp
```

If one of the columns is an expression, remember to use an alias. For example:

```sql
SELECT ename||' '||job display_value, empno FROM emp
```

**Note:** When creating a new item of this type, first select **Popup List of Values** and then **Text Field with Calculator Popup**.

Page Designer item attributes:

- **Settings, Input Field** - Select whether the item is enterable.
- **Settings, Fetch** - Select how data is initially fetched for the list of values.
- **Settings, Allow Multi Selection** - Allows for multiple, comma separated values to be selected.
- **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.

Component View item attributes:

- **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.
- **Settings, Allow Multi Selection** - Allows for multiple, comma separated values to be selected.

**See Also:** "Creating a Cascading List of Values" on page 15-10 and "Creating a Cascading List of Values" on page 15-20
### Table A–1 (Cont.) Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Radio Group | Renders as an HTML radio group form element, based on a list of values. The following example displays employee names (ename), but returns employee numbers (empno):  

```sql
SELECT ename, empno FROM emp
```  

Page Designer item attributes:  
- **Settings, Number of Columns** - Enter the number of radio group columns to display. For example, a value of 2 would display two columns. If there were five values then it would display over three rows.  
- **Settings, Page Action on Selection** - Select what action is taken when a radio group value is selected. See Help for more details.  
- **Settings, Execute Validations** - Displays if Page Action when Value Changed is set to **Submit Page**. Controls the execution of validations when a page is submitted. Set this attribute to **No** to prevent the execution of validations and associated error messages. Set this attribute to **Yes** to execute all page and built-in validations and associated error messages when the page is submitted. Use this attribute in conjunction with the validation attribute, **Always Execute**. To learn more, see "About Determining When Validations Execute" on page 17-15.  
  
**See Also**: "Creating a Cascading List of Values" on page 15-10 and "Creating a Cascading List of Values" on page 15-20  
- **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.  

Component View item attributes:  
- **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 radio group columns to display. For example, a value of 2 would display two columns. If there were five values then it would display over three rows.  
- **Settings, Page Action on Selection** - Select what action is taken when a radio group value is selected. 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.  
- **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" on page 17-15.  
  
**See Also**: "Creating a Cascading List of Values" on page 15-10 and "Creating a Cascading List of Values" on page 15-20
<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Rich Text Editor | Renders as an HTML text area. There is no maximum length for an item displayed as a text area. You control the height and width by editing the Height and Width item attribute. Page Designer item attributes:  
  - **Settings, Editor** - Select the editor used to display the text formatting options available.  
  - **Settings, Toolbar** - Select the number of text formatting options available within the editor.  
  - **Settings, Toolbar Expanded** - Determines if the toolbar must start expanded when the Rich Text Editor is loaded.  
  - **Settings, Skin** - Select the editor skin used to style the editor.  
  - **Settings, Toolbar Position** - Specify where in the text area the editor displays.  
  - **Validation, Value Required** - If select Yes and the page item is visible, Oracle Application Express automatically performs a NOT NULL validation when the page is submitted. See Help for more details. Component View item attributes:  
  - **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, Editor** - Select the editor used to display the text formatting options available.  
  - **Settings, Toolbar** - Select the number of text formatting options available within the editor.  
  - **Settings, Toolbar Expanded** - Determines if the toolbar must start expanded when the Rich Text Editor is loaded.  
  - **Settings, Skin** - Select the editor skin used to style the editor.  
  - **Settings, Toolbar Position** - Specify where in the text area the editor displays. |
Select List Displays using a list of values. A list of values is required for items displayed as a select list. Select lists are rendered using the HTML form element `<select>`. The values in a select list are determined using a named list of values or a list of values defined at the item level. You can specify the NULL display value and NULL return value.

The following example would return employee names (`ename`) and employee numbers (`empno`) from the `emp` table. Note that column aliases are not required and are included in this example for clarity.

```sql
SELECT ename display_text, empno return_value FROM emp
```

### Page Designer Item Attributes:
- **Page Action on Selection** - Select what action is taken when a list value is selected. Options include: None, Submit Page, Redirect and Set Value, Redirect to Page (based on selected value), or Redirect to URL (based on selected value).
- **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" on page 17-15.
- **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.

### Component View Item Attributes:
- **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.
- **Page Action on Selection** - Select what action is taken when a list value is selected. Options include: None, Submit Page, Redirect and Set Value, Redirect to Page (based on selected value), or Redirect to URL (based on selected value).
- **Allow Multi Selection** - Enables to select multiple values in the Select List. Use the Height attribute to define how many entries of the Select List should be visible at once.
- **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" on page 17-15.
### Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Shuttle   | Renders as a multiple select list that includes two boxes containing lists. The left list displays a source list of values. Users use the shuttle control icons and buttons to select list items and move them from the left (source) list to the right (destination) list. Each shuttle has five controls:  
  - Move all - Moves all items in the source list to the destination list.  
  - Move selected - Moves only selected items (Ctrl + Shift items) to destination list.  
  - Remove selected - Moves only selected items (Ctrl + Shift items) to the source list.  
  - Remove all - Moves all items on destination list back to the source list.  
  - Refresh - Resets the source and destination lists.  
  
  The right destination list includes the sort controls Move to top, Move up, Move down, and Move to bottom.  
  
  **Page Designer item attributes:**  
  - **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.  
  
  **Note:** To create this item type, you must define a list of values. See "Creating Lists of Values at the Application Level" on page 18-18, "Working with Multiple Select List Item" on page 15-4, and "Creating a Cascading List of Values" on page 15-10.  
  
  **Component View item attributes:**  
  - **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.  
  
  **Note:** To create this item type, you must define a list of values. See "Creating Lists of Values at the Application Level" on page 18-18, "Working with Multiple Select List Item" on page 15-4, and "Creating a Cascading List of Values" on page 15-20. |
Table A–1 (Cont.) Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slider</td>
<td>Renders slider item type for jQuery Mobile applications. This item type enables users to use slide handler to set a value.</td>
</tr>
</tbody>
</table>

Page Designer item attributes:

- **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.

Component View item attributes:

- **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

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Field</td>
<td>Displays the item as a text field.</td>
</tr>
</tbody>
</table>

Page Designer item attributes:

- **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.

- **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.

- **Settings, Submit when Enter Pressed** - Specify whether pressing the Enter key while in this field automatically submits the page.

- **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.

- **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.

Component View item attributes:

- **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, 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.

- **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.

- **Settings, Submit when Enter Pressed** - Specify whether pressing the Enter key while in this field automatically submits the page.

- **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

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Text Field with autocomplete| Displays a list of possible values when the user starts typing in the field. The values are based on the defined LOV SQL statement. The following example displays employee names (ename) and stores them in the field:  

SELECT ename FROM emp
ORDER BY 1

Page Designer item attributes:

- **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.

- **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.

Component View item attributes:

- **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.

- **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.
### Text Field with Calculator Popup

Renders as a text field with an icon. When clicked, the icon displays a small window containing a calculator. Calculations are placed back in the text field.

**Note:** When creating a new item of this type, first select **Popup List of Values** and then **Text Field with Calculator Popup**.

Component View item attributes:

- **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, Allow Multi Selection** - Allows for multiple, comma separated values to be selected.

### Text Area

Renders as an HTML text area. There is no maximum length for an item displayed as a text area. You control the height and width by editing the Height, Width, and Maximum Width attributes.

Page Designer item attributes:

- **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.
- **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.

Component View item attributes:

- **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, 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.

---

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Text Field with Calculator Popup | Renders as a text field with an icon. When clicked, the icon displays a small window containing a calculator. Calculations are placed back in the text field. **Note:** When creating a new item of this type, first select **Popup List of Values** and then **Text Field with Calculator Popup**. Component View item attributes:  
  - **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, Allow Multi Selection** - Allows for multiple, comma separated values to be selected. |
| Text Area                 | Renders as an HTML text area. There is no maximum length for an item displayed as a text area. You control the height and width by editing the Height, Width, and Maximum Width attributes. Page Designer item attributes:  
  - **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.  
  - **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. Component View item attributes:  
  - **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, 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. |
Table A–1  (Cont.) Available Item Types

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes/No</td>
<td>Displays Flip Toggle Switch in jQuery Mobile user interfaces and as a select list in non-mobile environments.</td>
</tr>
</tbody>
</table>

Component View item attributes:
- **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 following attributes display: Yes Value; Yes Label; No Value; No Label. To learn more, see Help.

Page Designer item attributes:
- **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 following attributes display: Yes Value; Yes Label; No Value; No Label. To learn more, see Help.

- **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.
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, "Component Limits"* describes limits associated with Oracle Application Express components.

**Component Limits**

*Table B–1, "Component Limits"* describes limits associated with Oracle Application Express components, such as interactive reports, classic reports, forms, and tabular forms.

<table>
<thead>
<tr>
<th>Component</th>
<th>Limits</th>
</tr>
</thead>
</table>
| Interactive reports | You can edit 100 columns using Component View. You can edit additional columns using Page Designer.  
999 rows per column heading filter (if no custom LOV is specified in the column attributes).  
You can edit additional columns by paginating through Report Column Attributes.  
**Note:** Depending on the column alias lengths, approximately 26 columns (or less) are supported.  
[See Also: "Developing Reports" on page 10-1](#)                                                                                       |
| Classic reports | 100 columns can be seen using report columns. You can edit additional columns using Page Designer.  
[See Also: "Developing Reports" on page 10-1](#)                                                                                       |
| Forms          | 200 enterable items per page.  
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" on page 11-1](#)                                                                                       |
| Tabular Forms  | One wizard-generated tabular form per page (using built-in DML).  
50 editable tabular form columns (apex_application.g_f01-apex_application.g_f50), generated with apex_item or the built-in tabular form display types.  
[See Also: "Managing Tabular Forms" on page 11-6](#)                                                                                       |
**Table B–1 (Cont.) Component Limits**

<table>
<thead>
<tr>
<th>Component</th>
<th>Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item names</td>
<td>Item names should not exceed 30 characters. Items longer than 30 characters cannot be referenced using bind variable syntax. See Also: &quot;Referencing Session State Using Bind Variable Syntax&quot; on page 2-24 and &quot;About Item Naming Conventions&quot; on page 2</td>
</tr>
<tr>
<td>Validations</td>
<td>Text entered for validations may not exceed 3,950 characters. See Also: &quot;Understanding Validations&quot; on page 17-6</td>
</tr>
<tr>
<td>Data loading</td>
<td>When loading data, the maximum number of columns is 45. See Also: &quot;Creating Applications with Data Loading Capability&quot; on page 25-1</td>
</tr>
</tbody>
</table>
Available Conditions

This section describes conditions available in Application Builder.
- About Conditions
- Conditions Available in Application Builder

About Conditions

A condition is a small unit of logic that helps you control the display of regions, items, buttons, and tabs as well as 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" on page 2-12 and "Conditions Available in Application Builder" on page C-1

Conditions Available in Application Builder

Table C–1, "Available Conditions" describes many Application Builder conditions. To view a complete listing of all available conditions for a given control or component, click the View icon to the right of the Condition Type list. Shortcuts to common selections appear directly beneath the Type list. If your condition requires an expression, type it in the appropriate field.

<table>
<thead>
<tr>
<th>Table C–1 Available Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
</tr>
<tr>
<td>Always</td>
</tr>
<tr>
<td>Client Browser: Microsoft Internet Explorer 5.5, 6.0 or later</td>
</tr>
<tr>
<td>Client Browser: Mozilla, Netscape 6.x/7.x or later</td>
</tr>
</tbody>
</table>
### Table C–1 (Cont.) Available Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Browser: Other browsers (or older version)</td>
<td>Evaluates to true when the client browser is one of the following: Other browsers (or older versions).</td>
</tr>
<tr>
<td>Client Browser: XHTML / CSS capable browser (NS 6.x,7x, Mozilla, IE 5.5, 6.0 or later)</td>
<td>Evaluates to true when the client browser is one of the following: Netscape 6.x, 7x, Mozilla, or Microsoft Internet Explorer 5.5, 6.0, or later). Setting this condition on an item hides the item for all other browsers other than Netscape 6.x, 7x, Mozilla, or Microsoft Internet Explorer 5.5, 6.0 or later).</td>
</tr>
<tr>
<td>Current Language != Expression 1</td>
<td>Verifies the language setting in which the client browser is not currently running. Checks that the current language is not the one entered in Expression 1.</td>
</tr>
<tr>
<td>Current Language = Expression</td>
<td>Verifies the language setting in which the client browser is currently running. Checks that the current language is not the one entered in Expression 1.</td>
</tr>
<tr>
<td>Current Language is contained within Expression 1</td>
<td>Determines whether the browser current language is contained within a string. Evaluates to true if the current language matches the string entered in Expression 1. For example, to check if the current language is either en-US or en-GB, choose this condition and enter the following string in Expression 1: en-us, en-gb</td>
</tr>
<tr>
<td>Current Language is not contained within Expression 1</td>
<td>Verifies the application's current language is not contained within a specified string. Evaluates to true if the current language is not contained within the string entered in Expression 1.</td>
</tr>
<tr>
<td>Current page != Expression 1</td>
<td>Evaluates to true if the current page does not equal the page you specify in Expression 1.</td>
</tr>
<tr>
<td>Current Page != Page Submitted (this page was not the page posted)</td>
<td>Determines if the specified page was not posted. Evaluates to true if the current page does not match the value entered in Expression 1.</td>
</tr>
<tr>
<td>Current page = Expression 1</td>
<td>Verifies whether the specified page was posted. Evaluates to true if the current page matches the value entered in Expression 1.</td>
</tr>
<tr>
<td>Current Page is contained within Expression 1 (comma-delimited list of pages)</td>
<td>Verifies if the current page is part of the list of pages you specify in Expression 1. To check if the current page is in either page 1, 2, 3 or 4, select this condition type and enter the following string in Expression 1: 1,2,3,4</td>
</tr>
<tr>
<td>Current page is in Printer Friendly mode</td>
<td>Only displays certain page control or components when the user has selected printer friendly mode. If the current page is in printer friendly mode, then the condition evaluates to true. Use <code>f?p</code> syntax to specify printer friendly mode.</td>
</tr>
<tr>
<td>Current page is not in Printer Friendly mode</td>
<td>Hides page controls or components when printer friendly mode is selected. Use <code>f?p</code> syntax to specify printer friendly mode. See Also: ”About Using f?p Syntax to Link Pages” on page 2-17 for information about <code>f?p</code> syntax</td>
</tr>
<tr>
<td>Current Page not in Expression 1 (comma-delimited list of pages)</td>
<td>Verifies if the current page is not part of the comma-separated list of pages specified in Expression 1.</td>
</tr>
<tr>
<td>Exists (SQL query returns at least one row)</td>
<td>This condition 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 emp where deptno = :P101_DEPTNO</td>
</tr>
<tr>
<td>Inline Validation Errors Displayed</td>
<td>This condition type hides page controls or components when inline validation errors are displayed on the page.</td>
</tr>
<tr>
<td>Never</td>
<td>This condition type is hard wired to always fail. It is useful in temporarily preventing controls or components (such as regions, buttons, or items) from being rendered on a page, or to prevent processes, computations and validations from running.</td>
</tr>
</tbody>
</table>
### Table C–1 (Cont.) Available Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Inline Validation Errors Displayed</td>
<td>This condition type hides page controls or components when no inline validation errors are displayed on the page.</td>
</tr>
<tr>
<td>NOT Exists (SQL query returns no rows)</td>
<td>This condition is expressed as a SQL query. If the query does not return any rows, it evaluates as true.</td>
</tr>
<tr>
<td>PL/SQL Expression</td>
<td>A PL/SQL expression is any expression in valid PL/SQL syntax that evaluates to true or false. For example: <code>nvl(:MY_FLOW_ITEM,'NO') = 'YES'</code> If the value of <code>MY_FLOW_ITEM</code> is YES then the condition evaluates as true. Otherwise it evaluates as false.</td>
</tr>
<tr>
<td>PL/SQL Function Body returning a Boolean</td>
<td>The body of a PL/SQL function that returns true or false. For example: <code>BEGIN IF :P1_DAY = 'MONDAY' THEN RETURN TRUE; ELSE RETURN FALSE; END IF; END;</code></td>
</tr>
<tr>
<td>Request != Expression 1</td>
<td>REQUEST is an internal attribute that tracks how a page is submitted. By default, when a page is submitted, the value of the application attribute REQUEST is set according to the name of the object that caused the page to be submitted. For a regular button, REQUEST is set as the name of the button (such as CANCEL or SAVE) not the label of the button. You can also set request using f?p syntax. For example, the event could be when a user clicks a button or selects a tab menu. Depending upon the event, you can perform different operations by referencing the value of the REQUEST application attribute. This condition evaluates as true if REQUEST does not equal the value entered in Expression 1. <strong>See Also:</strong> &quot;Understanding URL Syntax&quot; on page 2-16, &quot;REQUEST&quot; on page 2-38, and &quot;About the Relationship Between Button Names and REQUEST&quot; on page 15-49</td>
</tr>
<tr>
<td>Request = Expression 1</td>
<td>This condition is the opposite of Request != Expression 1. This condition evaluates as true if REQUEST equals the value entered in Expression 1. From PL/SQL you can also reference the application attribute using the following syntax: <code>V('REQUEST')</code> <strong>See Also:</strong> &quot;Understanding URL Syntax&quot; on page 2-16, &quot;REQUEST&quot; on page 2-38, and &quot;About the Relationship Between Button Names and REQUEST&quot; on page 15-49</td>
</tr>
<tr>
<td>Request is contained within Expression 1</td>
<td>REQUEST is an internal application attribute that tracks how a page is submitted. By default, when a page is submitted, the value of REQUEST is set according to the event that caused the page to be submitted. For example, the event could be when a user clicks a button or selects a tab. Depending upon the event, you can perform different operations by referencing the value of the REQUEST application attribute. Use this condition to specify a comma delimited list of allowed requests (such as SAVE or UPDATE) in Expression 1. The condition evaluates to true if the value of REQUEST is contained in the list. <strong>See Also:</strong> &quot;REQUEST&quot; on page 2-38 and &quot;About the Relationship Between Button Names and REQUEST&quot; on page 15-49</td>
</tr>
<tr>
<td>Request is not contained within Expression 1</td>
<td>This condition is the opposite of Request is contained within Expression 1. Evaluates to true if the value of the REQUEST is not contained within Expression 1. <strong>See Also:</strong> &quot;REQUEST&quot; on page 2-38 and &quot;About the Relationship Between Button Names and REQUEST&quot; on page 15-49</td>
</tr>
</tbody>
</table>
### SQL Expression
SQL Expressions are evaluated as a WHERE clause in a SQL statement. For example suppose your expression is `:MY_ITEM = 'ABC'`. The Application Express engine processes the following:
```sql
select 1 from dual where :MY_ITEM = 'ABC'
```
This condition evaluates to true if a row is returned.

### SQL Reports (OK to show the back button)
Use this condition for reports having pagination. It automatically determines when it is appropriate to include a button that pages back in the result set.

### SQL Reports (OK to show the forward button)
Use this condition for reports having pagination. It automatically determines when it is appropriate to include a button that pages forward in the result set.

### Text in Expression 1 != Expression 2 (includes &ITEM. substitutions)
Use this expression to compare two expressions containing strings. Either expression may contain references to session state using `&MY_ITEM.` syntax.

**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax

### Text in Expression 1 = Expression 2 (includes &ITEM. substitutions)
This condition is the opposite of `Text in Expression 1 != Expression 2` (includes `&ITEM. substitutions`). Compares two expressions containing strings. Either expression may contain references to session state using the `&ITEM.` syntax.

To check if the item `F100_P2_DAY_DATE` equals "Wednesday", select this condition enter the following in Expression 1 and Expression 2:

- Expression 1: `F100_P2_DAY_DATE`
- Expression 2: "Wednesday"

**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax

### Text in Expression 1 Is Contained in Value of Item in Expression 2
This condition compares the string contained in Expression 1 with the value of an item referenced in Expression 2. Expression 1 may contain references to session state using the `&ITEM.` syntax.

To check if the item `F100_P2_DAY_DATE` contains the text `Wed`, select this condition enter the following in Expression 1 and Expression 2:

- Expression 1: `Wed`
- Expression 2: `F100_P2_DAY_DATE`

**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax

### Text in Expression 1 Is Contained within the Text in Expression 2
This condition compares two expressions containing strings. Either expression may contain references to session state using the `&ITEM.` syntax.

**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax

### Text in Expression 1 Is NOT Contained within the Text in Expression 2
This condition is the opposite of `Text in Expression 1 Is Contained within the Text in Expression 2`. Either expression may contain references to session state using the `&ITEM.` syntax.

**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax

### User is authenticated (not public)
Verifies whether the current user was authenticated using one of the built-in authentication schemes or a custom authentication scheme.

**See Also:** "Establishing User Identity Through Authentication" on page 20-37 for information about authentication

### User is the public user (user has not authenticated)
The public user is defined as an application attribute. To set the public user for a specific application, navigate to the Application Builder home page and click the edit link corresponding to your application.

A public user is a user used for multiple users. Sometimes applications have pages that are public and thus require authentication and log in. This condition returns true if the user is the public user (that is, the user is authenticated as themselves or some other user not equal to the public user identified in the application attribute Public User.

**See Also:** "Authentication" on page 7-20 for information about Public User

---

### Table C–1 (Cont.) Available Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
</table>
| SQL Expression | SQL Expressions are evaluated as a WHERE clause in a SQL statement. For example suppose your expression is `:MY_ITEM = 'ABC'`. The Application Express engine processes the following:  
```sql
select 1 from dual where :MY_ITEM = 'ABC'
```
This condition evaluates to true if a row is returned. |
| SQL Reports (OK to show the back button) | Use this condition for reports having pagination. It automatically determines when it is appropriate to include a button that pages back in the result set. |
| SQL Reports (OK to show the forward button) | Use this condition for reports having pagination. It automatically determines when it is appropriate to include a button that pages forward in the result set. |
| Text in Expression 1 != Expression 2 (includes &ITEM. substitutions) | Use this expression to compare two expressions containing strings. Either expression may contain references to session state using `&MY_ITEM.` syntax. **See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax |
| Text in Expression 1 = Expression 2 (includes &ITEM. substitutions) | This condition is the opposite of `Text in Expression 1 != Expression 2` (includes `&ITEM. substitutions`). Compares two expressions containing strings. Either expression may contain references to session state using the `&ITEM.` syntax.  
To check if the item `F100_P2_DAY_DATE` equals "Wednesday", select this condition enter the following in Expression 1 and Expression 2:  
- Expression 1: `F100_P2_DAY_DATE`  
- Expression 2: "Wednesday"  
**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax |
| Text in Expression 1 Is Contained in Value of Item in Expression 2 | This condition compares the string contained in Expression 1 with the value of an item referenced in Expression 2. Expression 1 may contain references to session state using the `&ITEM.` syntax.  
To check if the item `F100_P2_DAY_DATE` contains the text `Wed`, select this condition enter the following in Expression 1 and Expression :  
- Expression 1: `Wed`  
- Expression 2: `F100_P2_DAY_DATE`  
**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax |
| Text in Expression 1 Is Not Contained within the Text in Expression 2 | This condition compares two expressions containing strings. Either expression may contain references to session state using the `&ITEM.` syntax.  
**See Also:** "Understanding Substitution Strings" on page 2-25 for information about `&MY_ITEM.` syntax |
| User is authenticated (not public) | Verifies whether the current user was authenticated using one of the built-in authentication schemes or a custom authentication scheme.  
**See Also:** "Establishing User Identity Through Authentication" on page 20-37 for information about authentication |
| User is the public user (user has not authenticated) | The public user is defined as an application attribute. To set the public user for a specific application, navigate to the Application Builder home page and click the edit link corresponding to your application.  
A public user is a user used for multiple users. Sometimes applications have pages that are public and thus require authentication and log in. This condition returns true if the user is the public user (that is, the user is authenticated as themselves or some other user not equal to the public user identified in the application attribute Public User.  
**See Also:** "Authentication" on page 7-20 for information about Public User |
**Table C–1 (Cont.) Available Conditions**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value of Item in Expression 1 != Expression 2</td>
<td>This condition is the opposite of Value of item in Expression 1 = Expression 2. It compares the value of an item with a specific string. Comparisons using this condition are case-sensitive. For example, to verify whether the value of an item F100_ID is not equal to the value 10, enter the following in the Expression 1 and Expression 2 fields:</td>
</tr>
<tr>
<td>Value of item in Expression 1 = Expression 2</td>
<td>Compares the value of an item with a specific string. Comparisons using this condition are case-sensitive. For example, to verify whether the value of an item F100_P2_WORD is contained within the string &quot;the quick brown fox&quot;, enter the following in the Expression 1 and Expression 2 fields:</td>
</tr>
<tr>
<td>Value of Item in Expression 1 = zero</td>
<td>Verifies if the value of the item in Expression 1 does equal zero.</td>
</tr>
<tr>
<td>Value of item in Expression 1 contains no spaces</td>
<td>Evaluate to true if the value of the item specified in Expression 1 contains no spaces.</td>
</tr>
<tr>
<td>Value of Item in Expression 1 is alphanumeric</td>
<td>Evaluates to true when the string in Expression 1 contains only alphanumeric characters.</td>
</tr>
<tr>
<td>Value of Item in Expression 1 is contained within colon-delimited list in Expression 2</td>
<td>Use this condition type to check whether a certain string is contained within the value of a session state item. Verifies whether the string specified in Expression 1 is contained in the value of the item specified in Expression 2.</td>
</tr>
<tr>
<td>Value of Item in Expression 1 is NOT contained within colon-delimited list in Expression 2</td>
<td>Evaluates to true when the value specified in Expression 1 contains a string that lists elements delimited by colons. To check if the item P1_TODAY is either &quot;Monday&quot;, &quot;Tuesday&quot;, or &quot;Wednesday&quot;, select this condition and enter the following in Expression 1 and Expression 2:</td>
</tr>
<tr>
<td>Value of Item in Expression 1 is NOT NULL</td>
<td>In Expression 1, enter the name (uppercase) of the application or page item. Evaluates as true, if the current cache value of the item is not null and has a value. If not, the condition evaluates as false.</td>
</tr>
<tr>
<td>Value of Item in Expression 1 Is NOT null and the Item Is NOT Zero</td>
<td>Use this condition to check whether the value in Expression 1 is not NULL and does not equal zero. For example, to verify whether the value of an item F100_ID is not null and is not equal to zero, enter the following in the Expression 1:</td>
</tr>
<tr>
<td>Value of Item in Expression 1 Is Not Numeric</td>
<td>Use this condition to verify that the value in Expression 1 is not numeric. For example, to verify whether the value of an item F100_USER_NAME is not numeric, enter the following in the Expression 1:</td>
</tr>
<tr>
<td>Value of Item in Expression 1 is NULL</td>
<td>Evaluates as true if the item in Expression 1 has no value.</td>
</tr>
<tr>
<td>Value of Item in Expression 1 is NULL or zero</td>
<td>Evaluates as true if the item in Expression is either NULL or zero.</td>
</tr>
<tr>
<td>Value of item in Expression 1 is numeric</td>
<td>Evaluates to true if the value of the Item in Expression 1 is numeric.</td>
</tr>
<tr>
<td>Value of user preference in Expression 1 != Expression 2</td>
<td>This condition is the opposite of Value of user preference in Expression 1 = Expression 2. Evaluates to true if the name of the user preference specified in Expression 1 is not equal to the string in Expression 2.</td>
</tr>
<tr>
<td>Value of user preference in Expression 1 = Expression 2</td>
<td>Verifies the value of a user preferences. Evaluates to true if the name of the user preference specified in Expression 1 is equal to the string in Expression 2.</td>
</tr>
<tr>
<td>When any item in comma-delimited list of items has changed</td>
<td>Evaluates to true when the value of any nonnull session state item in the list of items specified in Expression 1 has changed.</td>
</tr>
<tr>
<td>When any item in comma-delimited list of pages has changed</td>
<td>Evaluates to true when the value of any nonnull session state item in the list of pages specified in Expression 1 has changed.</td>
</tr>
</tbody>
</table>
### Conditions Available in Application Builder

#### Table C–1 (Cont.) Available Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When any item in current application has changed</td>
<td>This condition passes when the value of any nonnull session state item in the current application has changed.</td>
</tr>
<tr>
<td>When any item in current page has changed</td>
<td>Evaluate to true when the value of any nonnull session state item in the current page has changed.</td>
</tr>
<tr>
<td>When any item in current session has changed</td>
<td>Evaluates to true when the value of any nonnull session state item in the current session has changed.</td>
</tr>
<tr>
<td>When <code>cgi_env</code> <code>DAD_NAME</code> != Expression 1</td>
<td>This condition is the opposite of <code>When </code>cgi_env<code> </code>DAD_NAME<code> = Expression 1</code>. Checks for the data access descriptor (DAD) that is being used in the URL to call the current page in the application and compares it to Expression 1. Evaluate to true, when the DAD is not the same as Expression 1.</td>
</tr>
<tr>
<td>When <code>cgi_env</code> <code>DAD_NAME</code> = Expression 1</td>
<td>Checks for the data access descriptor (DAD) that is being used in the URL to call the current page in the application and compares it to Expression 1. Evaluate to true, when the DAD is the same as Expression 1.</td>
</tr>
<tr>
<td>When <code>cgi_env</code> <code>HTTP_HOST</code> != Expression 1</td>
<td>This condition is the opposite of <code>When </code>cgi_env<code> </code>HTTP_HOST<code> = Expression 1</code>. Checks for the value of the common gateway interface (CGI) environment variable <code>HTTP_HOST</code> that is the value returned by <code>owa_util.get_cgi_env('HTTP_HOST')</code>. Evaluate to true, when this value is not equal to the string in Expression 1.</td>
</tr>
<tr>
<td>When <code>cgi_env</code> <code>HTTP_HOST</code> = Expression 1</td>
<td>Checks for the value of the common gateway interface (CGI) environment variable <code>HTTP_HOST</code> that is the value returned by <code>owa_util.get_cgi_env('HTTP_HOST')</code>. Evaluate to true, when this value is equal to the string in Expression 1.</td>
</tr>
<tr>
<td>When <code>cgi_env</code> <code>SERVER_NAME</code> != Expression 1</td>
<td>This condition is the opposite of <code>When </code>cgi_env<code> </code>SERVER_NAME<code> = Expression 1</code>. This condition checks for the value of the common gateway interface (CGI) environment variable <code>SERVER_NAME</code>, that is the value returned by <code>owa_util.get_cgi_env('SERVER_NAME')</code>. Evaluate to true, when this value is not equal to the string in Expression 1.</td>
</tr>
<tr>
<td>When <code>cgi_env</code> <code>SERVER_NAME</code> = Expression 1</td>
<td>This condition checks for the value of the common gateway interface (CGI) environment variable <code>SERVER_NAME</code>, that is the value returned by <code>owa_util.get_cgi_env('SERVER_NAME')</code>. Evaluate to true, when this value is equal to the string in Expression 1.</td>
</tr>
</tbody>
</table>
About Granted Privileges

This section describes grant privileges.

- About Granting Privileges to Database Users
- Understanding Privileges Granted to PUBLIC

About Granting Privileges to Database Users

In Oracle Application Express release 4.2, no privileges are granted to database users when creating a workspace on an existing schema. Unless direct grants exist, database users in Oracle Application Express will not be able to create tables or other database objects. To allow the creation of typical database objects, Oracle recommends the following list of directly granted privileges:

```
grant CREATE SESSION to [database username]
grant CREATE CLUSTER to [database username]
grant CREATE DIMENSION to [database username]
grant CREATE INDEXTYPE to [database username]
grant CREATE JOB to [database username]
grant CREATE MATERIALIZED VIEW to [database username]
grant CREATE OPERATOR to [database username]
grant CREATE PROCEDURE to [database username]
grant CREATE SEQUENCE to [database username]
grant CREATE SNAPSHOT to [database username]
grant CREATE SYNONYM to [database username]
grant CREATE TABLE to [database username]
grant CREATE TRIGGER to [database username]
grant CREATE TYPE to [database username]
grant CREATE VIEW to [database username]
```

Understanding Privileges Granted to PUBLIC

This section describes public synonyms that exist and for which the execute privilege is granted to PUBLIC for the packages, procedures, functions, tables, and views owned by the Oracle Application Express product schema, (for example, APEX_050000).
Understanding Privileges Granted to PUBLIC

- **Packages**
- **Procedures**
- **Functions**
- **Tables**
- **Views**
- **Sequences**
- **Types**

**Packages**

Public synonyms exist and execute privilege is granted to `PUBLIC` for the following packages owned by the Oracle Application Express product schema, (for example, `APEX_050000`):

- APEXWS
- APEX_MIG_PROJECTS_UPDATE
- APEX_UI_DEFAULT_UPDATE
- HTMLDB_CUSTOM_AUTH
- HTMLDB_ITEM
- HTMLDB_LANG
- HTMLDB_SITE_ADMIN_PRIVS
- WWV_FLOW_APPLICATION_INSTALL
- HTMLDBUTIL
- WWV_FLOW
- WWV_FLOW_API
- WWV_FLOW_AUDIT
- WWV_FLOW_CSS_API
- WWV_FLOWAPPLICATION_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
Execute privilege is granted to PUBLIC for the following packages owned by the Oracle Application Express product schema, (for example, APEX_050000):

- WWV_CALCULATOR
- WWV_FLOW_FLASH_CHART
- WWV_FLOW_GENERATE_DDL
- WWV_FLOW_IMAGE_GENERATOR
- WWV_FLOW_TREE_GLOBAL_VARS
Procedures

Public synonyms exist and execute privilege is granted to PUBLIC for the following procedures owned by the Oracle Application Express product schema, (for example, APEX_050000):

APEX
APEX_ADMIN
DEVELOPMENT_SERVICE_HOME
DEVELOPMENT_SERVICE_HOME_LOGIN
DEVELOPMENT_SERVICE_SIGNUP
F
HTMLDB
HTMLDB_ADMIN
HTMLDB_LOGIN
P
WS
WWV_FLOW_INIT_HTP_BUFFER
Z

Functions

Public synonyms exist and execute privilege is granted to PUBLIC for the following functions owned by the Oracle Application Express product schema, (for example, APEX_050000):

NV
V

Execute privilege is granted to PUBLIC for the following function owned by the Oracle Application Express product schema, (for example, APEX_050000):

WWV_POPUP_FILTER

Tables

Public synonyms exist and select privilege is granted to PUBLIC for the following tables owned by the Oracle Application Express product schema, (for example, APEX_050000):

WWV_FLOW_DUAL100
WWV_FLOW_TEMP_TABLE
WWV_FLOW_LOV_TEMP

Views

Public synonyms exist and select privilege is granted to PUBLIC for the following views owned by the Oracle Application Express product schema, (for example, APEX_050000):

APEXAPPLICATIONS
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_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_FLASH_CH
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_CHK
APEX_APPLICATION_SUPP_OBJ_SCR
APEX_APPLICATION_TABS
APEX_APPLICATIONTEMPLATES
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_APL PLUGINS
APEX_APL PLUGIN_ATTRIBUTES
APEX_APL PLUGIN_ATTR VALUES
APEX_APL 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_PROJECTS
APEX_MIGRATION_REV_QUERIES
APEX_MIGRATION_REV_RPTS
APEX_MIGRATION_REV_TABLES
APEXRELEASE
APEX_STANDARD_CONDITIONS
APEX_TEAM_BUGS
APEX_TEAM_FEEDBACK
APEX_TEAM_MILESTONES
APEX_TEAM_TODOS
APEX_THEMES
APEX_UI_DEFAULTS_ATTR_DICT
APEX_UI_DEFAULTS_COLUMNS
APEX_UI_DEFAULTS_GROUPS
APEX_UI_DEFAULTS_LOV_DATA
APEX_UI_DEFAULTS_TABLES
APEX_WORKSPACES
APEX_WORKSPACE_ACCESS_LOG
APEX_WORKSPACE_ACTIVITY_LOG
APEX_WORKSPACE_APEX_USERS
APEX_WORKSPACE_CLICKS
APEX_WORKSPACE_DEVELOPERS
APEX_WORKSPACE_FILES
APEX_WORKSPACE_LOG_SUMMARY
APEX_WORKSPACE_LOG_SUMMARY_USR
APEX_WORKSPACE_SCHEMAS
APEX_WORKSPACE_SESSIONS
APEX_WORKSPACE_SQL_SCRIPTS
APEX_WS_APPLICATIONS
APEX_WS_APP_PAGES
APEX_WS_DATA_GRID
APEX_WS_DATA_GRID_COL
WWV_FLOW_CLICKTHRU_LOG
WWV_FLOW_COLLECTIONS
WWV_FLOW_FEEDBACK_TYPES
WWV_FLOW_GROUP_USERS
WWV_FLOW_LISTS_OF_VALUES$V
WWV_FLOW_LIST_OF_VALUES_DATA_V
WWV_FLOW_MONTHS_MON
WWV_FLOW_MONTHS_MONTH
WWV_FLOW_PLSQL_JOBS
WWV_FLOW_USERS
WWV_FLOW_USER_ACCESS_LOG
WWV_FLOW_USER_ACTIVITY_LOG
WWV_FLOW_USER_MAIL_LOG
WWV_FLOW_YEARS

Additionally, insert is granted on the following:

WWV_FLOW_FILES

Additionally, the following views have select privilege granted to PUBLIC, but no synonym:

WWV_FLOW_HOURS_12
WWV_FLOW_HOURS_24
WWV_FLOW_MINUTES
WWV_FLOW_MINUTES_5

Sequences

The following sequences have public synonyms and select privilege granted to public:

WWV_FLOW_SESSION_SEQ
WWV_SEQ

Types

The following types have execute granted to public but no synonyms:

WWV_FLOW_TREE_ENTRY
WMV_FLOW_TREE_SUBS
Accessibility in Oracle Application Express

This section provide information for users who are accessing Oracle Application Express utilizing only a keyboard or Freedom Scientific's screen reader JAWS.

**Note:** JAWS release 16.0 with all default settings was used in writing this section. All instructions in this section are based on using a full size keyboard. Note that instructions may vary if using a laptop keyboard.

- Understanding Screen Reader Mode
- Understanding High Contrast Mode
- Accessing Form Pages Using JAWS
- Accessing Oracle Application Express Using a Keyboard

**See Also:** Oracle’s Accessibility Program for additional information about the accessibility of Oracle products:


### Understanding Screen Reader Mode

This section describes screen reader mode Sessions can be defined as optimized for use with screen readers. This includes both when using the Application Express development environment and also for use in your own applications.

- Enabling Screen Reader Mode
- Using Screen Reader Mode in Your Database Applications
- What Does Screen Reader Mode Do?

### Enabling Screen Reader Mode

You can enable screen reader mode by selecting **Screen Reader** from Accessibility Mode list on the Workspace home page and on the Oracle Application Express Administration Services page.
You can enable screen reader mode in the Websheet application by clicking the **Screen Reader** link located in the lower right side of the Websheet home page.

If you intend on using the Oracle Application Express development environment with a screen reader, ensure that you have this option enabled.

**Using Screen Reader Mode in Your Database Applications**

Sessions in Oracle Application Express can now be identified as optimized for screen readers, in the Oracle Application Express development environment, Websheet runtime, and also within your own database applications. By default sessions are not flagged as running in screen reader mode. This section describes how you can provide users of your own database applications access to this mode and also how you can extend it.

**Tip:** Although screen reader mode improves the usability of Oracle Application Express with a screen reader, there may still be some areas where outstanding issues remain. For a full list of outstanding issues with workarounds where possible, see "What Does Screen Reader Mode Do?" on page E-3 and Oracle Application Express Release Notes.

- Provisioning Screen Reader Mode
- Extending Screen Reader Mode

**See Also:** "Understanding Screen Reader Mode" on page E-1

**Provisioning Screen Reader Mode**

You can provision screen reader mode for use with your own database application in three ways:

1. Use the page template `#SCREEN_READER_TOGGLE#` substitution string.
   
   Add the `#SCREEN_READER_TOGGLE#` substitution string to your page template and Oracle Application Express displays a link to the current page to turn on or off (toggle) the mode. So if you are in standard mode, this procedure generates a link to turn it on.

2. Use APIs to control screen reader mode.
   
   You may want to use the APIs if you only want to render the toggle in one place and not at the page template level, or if you want more control over the actual displayed link text. To learn more, see "APEX_UTIL" in Oracle Application Express API Reference.

3. Use the f?p syntax REQUEST attribute to enable and disable screen reader mode.
   
   The general syntax for f?p is:

   ```
f?p=application:page:session:request:...
```

   If the request is exactly `SET_SESSION_SCREEN_READER_ON` or `SET_SESSION_SCREEN_READER_OFF` then the session is put into or out of Screen Reader mode. For example:

   ```
   <a href="f?p=100:1:&SESSION.:SET_SESSION_SCREEN_READER_ON">Set Screen Reader On</a>
   ```
Extending Screen Reader Mode

In addition to what this mode does by default as detailed in "What Does Screen Reader Mode Do?" on page E-3, you can control your own page components. For example, you can conditionally display region when running in Screen Reader mode. See "APEX_UTIL" in Oracle Application Express API Reference for details of the relevant database functions, APEX_UTIL.IS_SCREEN_READER_SESSION and APEX_UTIL.IS_SCREEN_READER_SESSION_YN.

What Does Screen Reader Mode Do?

Enabling Screen Reader mode currently does the following, by default:

- Oracle Application Express Flash charts are not currently accessible to screen readers, therefore when running in screen reader mode the user gets a report representation of the information conveyed in the chart. A separate report is generated for each series of a multiple-series chart. When running in screen reader mode, these data tables contain descriptive text, in the following format:
  - Database Applications:
    * Summary Text - A combination of the chart title and chart series title are used.
    * Column Headers - The column name/alias in the chart series query identifies the columns in the report.
  - Websheet Applications:
    * Summary Text - The section title is used as the summary text.
    * Column Headers - The Data Grid report labels identify the columns in the report.

- Interactive report regions in Report View or Group By View use data tables to convey information. Data tables require a text description that describes what information the table contains to a screen reader user. When running in screen reader mode, these data tables now contain detailed summary text, in the following format:
  - Region - The name of the region defined by the developer.
  - Report - The name of the current Saved Report within the Interactive Report. Default to Primary Default if no saved report is defined.
  - View - The current view of the report (Either Report or Group By).
  - Displayed Rows Start - The starting row of the currently displayed set of rows.
  - Displayed Rows End - The ending row of the currently displayed set of rows.
  - Total Rows - The total number of rows returned by the report.

- Interactive report regions provide the ability to add Aggregates to your report data. One of the available aggregates is to Sum a column's values. When running in screen reader mode, the text Sum: is displayed as a prefix to the summed value. In standard mode only the summed value is displayed. All other aggregates display an aggregate type prefix by default, regardless of mode.

- Item level Help text is provided differently in screen reader mode. In standard mode, a dialog is displayed on your page displaying the Help text for a specific item. Since these dialogs have many problems when used with screen readers, a the classic popup page displays item help text when running in screen reader mode.
Application Express includes two implementations of the Tree region component in database Applications: Tree and jsTree (legacy). For improved accessibility, Oracle recommends using Tree. jsTree (legacy) has some accessibility problems, and for this reason when running in screen reader mode the user gets an unordered hierarchical list as a representation of the information conveyed in the tree.

Screen reader mode has greatly improved the process for the user of dealing with raised validation errors. Consider the following example. A form page is submitted and a validation has failed resulting in displayed errors. If the page item has an associated error, then additional markup is added to that page item so that the screen reader informs the user that the item contains an invalid entry, when moving through the form fields. Additionally, the user is informed of the corresponding error message, when moving through the form fields.

**Tip:** You must be using both a screen reader and browser that supports WAI-ARIA, to benefit from these improvements.

Interactive report regions have a Fixed Header option that enables developers to fix the headers of the interactive report for visual users. Because this functionality changes the underlying table structure and has a detrimental effect for screen reader users, this option is disabled when running in Screen Reader mode.

---

**Understanding High Contrast Mode**

This section describe high contrast mode. You can identify sessions as optimized for high contrast in Websheet and Database applications. By default, sessions are not flagged as running in high contrast mode.

- **Enabling High Contrast Mode in a Websheet Application**
- **Using High Contrast Mode in Your Database Applications**
- **What Does High Contrast Mode Do?**

**Enabling High Contrast Mode in a Websheet Application**

You can enable High Contrast mode in a Websheet application by clicking the **High Contrast** link located in the lower right side of the Websheet home page.

**Using High Contrast Mode in Your Database Applications**

You can identify sessions as optimized for high contrast in Websheet applications and also within your own database applications. This section describes how you can provide users of your own database applications access to this mode and also how you can extend it.

- **Provisioning High Contrast Mode**
- **Extending High Contrast Mode**

**See Also:** "Understanding High Contrast Mode" on page E-4

**Provisioning High Contrast Mode**

You can provision high contrast mode for user's of your own database applications in three ways:

- Use the page template #HIGH_CONTRAST_TOGGLE# substitution string.
Add the `#HIGH_CONTRAST_TOGGLE#` substitution string to your page template and Oracle Application Express displays a link to the current page to turn on or off (or toggle) the mode. In other words, if you are in standard mode, this procedure generates a link to turn it on.

- Use APIs to control high contrast mode.
  
  Use the `APEX_UTIL` APIs to control this mode. You may want to use the APIs if you only want to render the toggle in one place and not at the page template level, or if you want more control over the actual displayed link text. To learn more, see "APEX_UTIL" in Oracle Application Express API Reference.

- Use the `f?p` syntax REQUEST attribute to enable and disable high contrast mode.
  
  The general syntax for the `f?p` syntax REQUEST attribute is:

  ```sql
  f?p=application:page:session:request:...
  ```

  If the request is exactly `SET_SESSION_HIGH_CONTRAST_ON` or `SET_SESSION_HIGH_CONTRAST_OFF` then the session is put into or out of high contrast mode. For example:

  ```html
  <a href="f?p=100:1:&SESSION.:SET_SESSION_HIGH_CONTRAST_ON">Set High Contrast On</a>
  ```

**Extending High Contrast Mode**

In addition to what this mode does by default as described in "What Does High Contrast Mode Do?" on page E-5, you can also control your own components. For example, you could conditionally include a high contrast specific CSS file, when running in high contrast mode.

To learn more, see "APEX_UTIL" in Oracle Application Express API Reference details of the following functions: `APEX_UTIL.IS_HIGH_CONTRAST_SESSION` and `APEX_UTIL.IS_HIGH_CONTRAST_SESSION_YN`.

**What Does High Contrast Mode Do?**

Enabling high contrast mode results in the following behavior:

- Application Express Websheet runtime.
  
  Websheet application replace CSS-based image links with text links. These text links are only created for images where the source image is transparent and the CSS background property is used in its place.

- Custom Database Applications.
  
  Nothing is done natively from the framework. It is up to each developer to provision the mode and make changes where required.

**Accessing Form Pages Using JAWS**

Many of the pages in Oracle Application Express are form pages designed for maintaining application metadata. This section describes how you can gain an understanding of the structure of these pages and interact with them using a screen reader such as JAWS.

- **About the Structure of Pages**
  
  - Accessing Help Text
Dealing with Validation Errors

About the Structure of Pages

Many form pages are logically broken up into different regions. For example, the Edit Page Item page has the following regions containing form fields for maintaining information about the page item:

- Name
- Displayed
- Label
- Settings
- List of Values (depending on item type)
- Element
- Source
- Default
- Quick Picks (depending on item type)
- Conditions
- Read Only
- Security
- Configuration
- Help Text
- Comments

Oracle Application Express uses Heading (Hn) tags throughout its development interface to define page structure. For example, the Edit Page Item page contains one H1 heading with text pertaining to the current item name, and then H2 headings for all other discrete page sections, including all the item attribute groupings listed above. Please use standard screen reader heading navigation to navigation such pages.

Accessing Help Text

Most items within the Application Express development environment have field-level help defined. To activate field-level help, focus on the item for which you need help and press the keyboard shortcut \textit{ALT+F1}. Pressing this keyboard shortcut displays a popup window containing the help text.

Dealing with Validation Errors

Oracle Application Express provides many in-built features that help a JAWS user deal validation errors on form pages.

- About Understanding the an Error Has Occurred
- Reviewing the Error
- Adding a Validation Notification to Your Database Applications
- Navigating to the Relevant Fields to Fix the Errors
- About WAI-ARIA Support
About Understanding the an Error Has Occurred
If a validation has failed on a form page when using Screen Reader mode, the page title is prefixed with the following:

x errors have occurred - [page title]

When a page loads, JAWS announces the page title. This means a JAWS user immediately knows a validation has failed.

**Tip:** This behavior is also available by default for custom database applications that provision Screen Reader mode. See "Understanding Screen Reader Mode" on page E-1.

Reviewing the Error
To navigate to the section of the page where the error messages display, locate the H2 (Heading Level 2) tag with the text Notification Message included at the top of this section. This heading is hidden visually, but is available to screen reader users.

Adding a Validation Notification to Your Database Applications
You can easily add a validation notification to your own database applications. To accomplish this, add the substitution string #NOTIFICATION_MESSAGE_HEADING# in the Page Template, Subtemplate, Notification section of a page template. This placeholder substitutes the contents of the built-in system message APEX.NOTIFICATION_MESSAGE_HEADING to the text Notification Message.

To customize this displayed text, define a text message with that name APEX.NOTIFICATION_MESSAGE_HEADING in your own application, along with the text you want to use.

For example, you could add the following to the Notification section of your page template:

```
<h2 class="hideMeButHearMe">#NOTIFICATION_MESSAGE_HEADING#</h2>
```

In this example, the use of the hideMeButHearMe CSS class hides content visually, but is available to screen reader users.

Navigating to the Relevant Fields to Fix the Errors
For validations that are associated with a particular page item, Oracle Application Express renders a Go to error link in the Notification section next to the corresponding error message. When a user clicks (or activates) the Go to error link, the focus shifts to the associated item. This enables all users (not just screen reader users) to go straight to the page item related to the error.

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" on page 21-14
About WAI-ARIA Support
When running in Screen Reader mode, Oracle Application Express includes some Web Accessibility Initiative - Accessible Rich Internet Applications (WAI-ARIA) markup to invalid page items. This enhancement improves the screen reader user experience for screen reader and browser combinations that support WAI-ARIA.

When an error occurs relating to a specific page item, that page item is given the following additional attributes:

- `aria-invalid="true"
  This means a screen reader announces "Invalid Entry" when the screen reader user is focused on the page item.

- `aria-describedby="[page_item_name]_error"
  This attribute value matches up with the item’s associated error messages so when the screen reader user focuses on the page item, the screen reader announces the actual error.

Before this enhancement, if a screen reader user navigated to a page item that had an error, the screen reader might announce, "Job, Edit, Type of Text." Now the screen reader would announce, "Job Edit, Invalid Entry, Job must have some value, Type of Text."

**Tip:** If you are a plug-in item developer and want to provide the same WAI-ARIA support in your own plug-in, please see the `APEX_ERROR.GET_ARIA_ERROR_ATTRIBUTES` function in Oracle Application Express API Reference.

Accessing Oracle Application Express Using a Keyboard
This section describes how to access components in Oracle Application Express using a keyboard.

- Using New Date Pickers
- Accessing Interactive Report Regions
- Keyboard Shortcuts

Using New Date Pickers
There are many keyboard shortcuts for the Date Picker.

To open a Date Picker:
1. Tab to the relevant input field.
2. Press **CTRL+HOME** to open the Date Picker.
3. Once the Date Picker is open, you have the following navigation options:
   - By month:
     - Press **PAGE UP** to go to the previous month.
     - Press **PAGE DOWN** to go to the next month.
     - Press **CTRL +HOME** to go to the current month.
   - By week:
     - Press **CTRL+UP** to go to the previous week.
– Press **CTRL+DOWN** to go to the next week.

  ■ **By day:**
    – Press **CTRL+LEFT** to go to the previous day.
    – Press **CTRL+RIGHT** to go to the next day.

Once you are finished with the Date Picker:

1. Press **ENTER** to accept the selected date.
2. Press **ESCAPE** to close the date picker without selection.
3. Press **CTRL+END** to close the date picker and erase the date.

**Accessing Interactive Report Regions**

This section describes how to access common features within Interactive Report regions.

- **Searching Data in an Interactive Report**
- **Using the Actions Menu**
- **Managing Bulk Updates**
- **Utilizing Drill Down Links**
- **Editing Existing Filters**
- **Loading Saved Reports**

**Searching Data in an Interactive Report**

If a search bar displays above an interactive report then the Include Search Bar attribute has been enabled in Report Attributes. This attribute is enabled by default when a new interactive report is created.

To execute a search:

1. Press **TAB** until the search field has focus.
2. Type your search keywords into the search field.
3. Press **ENTER** to invoke the search.
4. To refine your search to a specific column:
   a. Press **TAB** to access the magnifying glass icon to the left of the search field.
   b. Press **ENTER** to activate the pull down menu and display a list of all available columns to search.
   c. Press the **DOWN** or **UP** arrow keys on your keyboard to navigate through the list of **columns**.
   d. Press **ENTER** to select the column you want to search on. This sets the context of the subsequent search as specific to the column selected.
5. Repeat steps 1 and 2 to execute your column specific search.

**Using the Actions Menu**

By default, new interactive reports include both a search bar and Actions menu. Both of these features are controlled by the developer by enabling the Include Search Bar and Actions menu attributes on the Report Attributes page. Depending upon how the developer creates the report, the Action menu can contain different options.
To access the Actions menu using a keyboard:

1. Press TAB until the Actions menu has focus.
2. Press ENTER to open the menu.
3. Once the menu is open, you have the following navigation options:
   - Press the DOWN arrow key to go down the menu.
   - Press the UP arrow key to go up the menu.
   - Press the RIGHT arrow key to open a submenu (indicated by a right arrow) and move focus to the first item in the submenu.
   - Press the LEFT arrow key to close a submenu and move focus back to the appropriate item in the parent menu.
4. To exit the Action menu:
   - Press ENTER to load the action specific dialog. Focus is set to the first focusable element within the dialog box.
   - Press ESC to close the menu without selection.

Managing Bulk Updates

Some interactive reports in Oracle Application Express contain form items for applying bulk updates to components. All form items within the report and buttons to apply the changes are keyboard accessible. Also some pages feature a check box in the first row to perform bulk processing and corresponding Select All and Deselect All links below the report. All of these items are keyboard accessible.

Utilizing Drill Down Links

Some interactive reports in Oracle Application Express contain at least one link to edit or view a specific sub-components. These are all standard links and are therefore keyboard accessible.

Editing Existing Filters

When an existing search filter is applied to an interactive report, a search filter displays below the Search bar.

You can access the search filter using the keyboard to:

- Edit the search filter.
  - Press TAB to navigate to the search filter and then press ENTER to open the Filter dialog box. Focus is automatically set to first focusable element in the dialog box.
- Disable or enable the search filter.
  - Press TAB to navigate to the check box to the left of the filter and SPACEBAR to disable or enable the filter.
- Delete the search filter.
  - Press TAB to navigate to the search filter and then ENTER to open the Filter dialog box. Press TAB or SHIFT+TAB to navigate to the Delete button and then press ENTER.

See Also: "About the Actions Menu" on page 10-58
Loading Saved Reports
If an Interactive Report includes a Search bar, Actions menu, and support for saved reports, then these options have been enabled by the developer on the Report Attributes page.

See Also: "Editing Interactive Reports in Page Designer" on page 10-11 and "Editing Interactive Reports in Component View" on page 10-33

To load a saved report:
1. Press TAB until the Reports select list has focus. Note that the Reports select list only displays if there are saved reports.
2. Press the UP and DOWN keys to select the specific saved report you want to view.
3. Press ENTER to view the saved report.

Keyboard Shortcuts
Table E–1 lists available Oracle Application Express keyboard shortcuts.

<table>
<thead>
<tr>
<th>Where</th>
<th>Keys</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Designer Grid Layout</td>
<td>CTRL+ALT+D</td>
<td>Display from here</td>
</tr>
<tr>
<td>Page Designer Grid Layout</td>
<td>CTRL+ALT+T</td>
<td>Display from page</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+2</td>
<td>Go to Dynamic Actions</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+9</td>
<td>Go to Gallery Buttons</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+8</td>
<td>Go to Gallery Items</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+7</td>
<td>Go to Gallery Regions</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+5</td>
<td>Go to Grid Layout</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+F1</td>
<td>Go to Help</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Ctrl+F1</td>
<td>Go to Messages</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+4</td>
<td>Go to Page Shared Components</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+3</td>
<td>Go to Processing</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+6</td>
<td>Go to Property Editor</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+1</td>
<td>Go to Rendering</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+Shift+F1</td>
<td>List Keyboard Shortcuts</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Ctrl+Alt+F</td>
<td>Page Search</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Ctrl+Y</td>
<td>Redo</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Alt+F11</td>
<td>Restore/Expand</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Ctrl+Alt+S</td>
<td>Save</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Ctrl+Alt+R</td>
<td>Save and Run Page</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Ctrl+Alt+E</td>
<td>Toggle Hide Empty Positions</td>
</tr>
<tr>
<td>Page Designer</td>
<td>Ctrl+Z</td>
<td>Undo</td>
</tr>
<tr>
<td>Where</td>
<td>Keys</td>
<td>Action</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>Page Designer with focus anywhere that has a context menu</td>
<td>Shift+F10 or Context menu key</td>
<td>Open context menu</td>
</tr>
<tr>
<td>Page Designer Grid Layout Gallery and Icon List in wizards</td>
<td>Arrow keys</td>
<td>Move selection</td>
</tr>
<tr>
<td>Page Designer Property Editor with focus on a group heading</td>
<td>Up/Down Arrow keys</td>
<td>Move to previous/next group</td>
</tr>
<tr>
<td>Page Designer Property Editor with focus on a group heading</td>
<td>Home/End keys</td>
<td>Move to first/last group</td>
</tr>
<tr>
<td>Page Designer Property Editor with focus on a group heading</td>
<td>Left/Right Arrow keys or Enter or Space</td>
<td>Expand or collapse the group</td>
</tr>
<tr>
<td>Page Designer With focus on any tab</td>
<td>Arrow keys</td>
<td>Select previous/next tab</td>
</tr>
<tr>
<td>Focus on a splitter handle</td>
<td>Arrow keys</td>
<td>Move splitter</td>
</tr>
<tr>
<td>Focus on a splitter handle</td>
<td>Enter</td>
<td>Expand or collapse splitter if supported by splitter</td>
</tr>
<tr>
<td>On pages that have a splitter</td>
<td>Ctrl+F6</td>
<td>Move to next splitter</td>
</tr>
<tr>
<td>On pages that have a splitter</td>
<td>Ctrl+Shift+F6</td>
<td>Move to previous splitter</td>
</tr>
<tr>
<td>Focus on any field with a (?) help icon</td>
<td>Alt+F1</td>
<td>Open field help dialog</td>
</tr>
<tr>
<td>Focus on a field help dialog</td>
<td>Alt+F6</td>
<td>Move focus back to field without closing dialog</td>
</tr>
<tr>
<td>Focus in any dialog</td>
<td>Escape</td>
<td>Close dialog</td>
</tr>
<tr>
<td>Any page, focus on region display selector “tabs”</td>
<td>Arrow keys</td>
<td>Select the previous/next tab</td>
</tr>
<tr>
<td>SQL Commands</td>
<td>Ctrl+Enter</td>
<td>Run the current command</td>
</tr>
<tr>
<td>Code Editor</td>
<td>Ctrl+Z</td>
<td>Undo</td>
</tr>
<tr>
<td>Code Editor</td>
<td>Ctrl+Shift+Z</td>
<td>Redo</td>
</tr>
<tr>
<td>Code Editor</td>
<td>Ctrl+F</td>
<td>Find</td>
</tr>
<tr>
<td>Code Editor</td>
<td>Ctrl+Shift+F</td>
<td>Replace</td>
</tr>
<tr>
<td>Code Editor with focus in find field</td>
<td>Up/Down Arrows</td>
<td>Find previous/next instance</td>
</tr>
<tr>
<td>Code Editor</td>
<td>Ctrl+Space</td>
<td>Auto complete when available</td>
</tr>
<tr>
<td>Code Editor</td>
<td>Alt+F6</td>
<td>Leave code editor and go to next tab stop</td>
</tr>
<tr>
<td>Code Editor</td>
<td>Alt+Shift+F6</td>
<td>Leave code editor and go to previous tab stop</td>
</tr>
</tbody>
</table>
A

Accept Page, 2-12
access control list
  about, 20-32
  adding users, 20-35
  creating, 20-33
  removing users, 20-36
  selecting an application mode, 20-34
  selecting user privileges, 20-35
access control, authorization schemes, 20-36
accessibility
  about, E-1
  about high contrast mode, E-4
  Accessibility Mode list, E-1
  accessing components with keyboard, E-8
  screen reader mode, E-1
  WAI-ARIA support, E-8
Accessibility Mode list
  Screen Reader, E-1
Account menu, 1-16
account profile
  changing your password, 1-17
  editing email address, 1-17
  editing name, 1-17
  uploading a photo, 1-17
Actions menu, 10-58
  Aggregate, 10-58
  Chart, 10-58
  Compute, 10-58
  Control Break, 10-58
  customizing, 10-16
  Download, 10-59
  Filter, 10-58
  Flashback, 10-59
  Format, 10-58
  Group By, 10-58
  Help, 10-59
  Highlight, 10-58
  Reset, 10-59
  Rows Per Page, 10-58
  Save Report, 10-59
  Select Columns, 10-58
  Sort, 10-58
  Subscription, 10-59
activity log, 23-6
activity reports, creating custom, 23-6
Administration Menu
  Administration, 1-14
  Change My Password, 1-15
  Dashboards, 1-15
  Manage Service, 1-14
  Manage Users and Groups, 1-14
  Monitory Activity, 1-15
Administration menu, 1-14
Allow Feedback attribute, 7-16
Allows URLs Created After attribute, 7-24
AnyCo IT Department
  about, 3-17
  installing, 3-17
  navigating, 3-17
  running, 3-17
APEX_ACTIVITY_LOG view, 23-6
APEX_MAIL, 19-2
APEX_PUBLIC_USER, 23-5
application
  adding content from Web sites, 14-24
  adding Data Load Wizard, 25-1
  adding data loading, 25-1
  attaching an authorization scheme, 20-52
  changing authentication scheme, 20-47, 20-48
  controlling navigation using branches, 16-20
  copying, 4-8
  creating, 2-1
  debugging, 22-1, 23-1
  defining primary language, 21-4
  deleting, 4-8
  deploying, 24-3, 24-5, 24-14
  exporting, 24-5, 24-14
  grouping, 7-59
  hiding features, 24-32
  importing export file, 24-23
  importing from SQL*Plus, 24-29
  language preference, 21-2
  resource use, 22-3
  translatable components, 21-2
  translating, 21-1
  translating multibyte languages, 21-6
  translation rendering, 21-2
  upgrading, 7-63, 7-64
Application Alias attribute, 7-15
application attributes
Build Options, 7-19
Database Session, 7-25
ing, 7-14
Error Handling, 7-17
exporting, 24-16
Substitutions, 7-19
Application Builder
accessing, 4-1, 4-3
home page, 4-2
page, 2-4
utilities, 7-61
Application Builder home page, 4-1, 4-2, 4-3
Create button, 4-3
Migration link, 4-3
navigation bar, 4-2
Recent list, 4-3
Reset button, 4-3
Application Builder utilities
Advisor, 7-62
Application Dashboard, 7-62
Application Express Views, 7-62
Attribute Dictionary, 7-62
Change History, 7-62
Database Object Dependencies, 7-62
Debug Messages, 7-62
Export Repository, 7-62
Recently Updated Pages, 7-62
Upgrade Application, 7-62
application computations
about, 17-30
creating, 17-31
executing On New Instance, 17-26, 17-30
history report, 17-31
Application Dashboard, 7-62
Application Dashboard report, 23-3
Application Date Format attribute, 7-27
Application Date Time Format attribute, 7-27
application definition, editing, 7-14
Application Express Accounts authentication
setting up, 20-40
Application Express Views, 7-65
application groups, 7-59
Application Groups attribute, 7-15
application Help, creating, 12-39
Application home page
adding database application page, 7-40
Export/Install icon, 3-16, 4-5
Run Application button, 4-5
Run Application icon, 3-16
Shared Components icon, 3-16, 4-5
Application home Task list
Copy this Application, 4-8
Delete this Application, 4-8
application item
about, 15-1, 18-15
creating, 18-15
creating a global variable, 18-17
editing attributes, 18-17
history, 18-16
application language, 7-26
Application Language Derived From attribute, 7-27, 21-4
application layout
print preview mode, 14-2
shortcuts, 18-34
Application Primary Language attribute, 7-27, 21-4
application processes
creating, 17-27
creating error message, 17-28
editing attributes, 17-28
example, 17-27
On Demand, 17-26
reports, 17-29
understanding, 17-25
Application Timestamp Format, 7-28
Application Timestamp Time Zone Format, 7-28
application utilities
accessing, 7-61
architecture, 1-2
asynchronous updates
charts, 12-28, 12-31
maps, 12-19, 12-22
Attribute Dictionary, 17-32
about, 17-32
accessing, 17-32
authentication, 2-13, 20-37
Authentication Schemes Repository, 20-47
changing, 20-47, 20-48
creating an authentication scheme, 20-39
managing existing schemes, 20-47
preconfigured authentication schemes, 20-39
reports, 20-48
security attributes, 7-20
understanding, 20-37
viewing current scheme, 20-47
authentication report
Authentication Scheme History, 20-49
Authentication Scheme Subscription, 20-49
Authentication Scheme attribute, 7-20
Authentication Scheme History report, 20-49
Authentication Scheme Subscription report, 20-49
authentication schemes
Application Express Accounts, 20-39
cookie user accounts, 20-39
Custom authentication, 20-40
DAD Credentials, 20-44
Database Account Credentials, 20-42
HTTP Header Variable, 20-42
LDAP Directory Verification, 20-43
Open Door Credentials, 20-45
Single Sign-On verification, 20-46
authorization, 2-14
Authorization attributes, 7-21
authorization scheme
attaching to application, 20-52
attaching to component, 20-53
attaching to control, 20-53
attaching to page, 20-53
By Component report, 20-53
History report, 20-54
of an application, 7-21
reports, 20-53
specifying, 7-21
Subscription report, 20-53
Utilization report, 20-54
Authorization Scheme attribute, 7-21
Authorization Scheme History report, 20-54
Authorization Scheme Subscription report, 20-53
Authorization Scheme Utilization report, 20-54
authorization schemes, 20-49
attaching to application, 20-52
attaching to component, 20-53
attaching to control, 20-53
attaching to page, 20-53
creating, 20-49
for controlling access, 20-36
Automatic CSV Encoding attribute, 7-29
Automatic Time Zone, 7-28

B

bind variables
syntax, 2-24
using in regions, 2-24
using in regions based on PL/SQL, 2-25
blank page
creating with Create Application Wizard, 7-2
BLOB support in forms
about, 11-22
displaying the BLOB, 11-23
downloading files, 11-22
removing the image reference, 11-23
uploading files, 11-22
BLOB support in reports
about, 10-89
displaying the BLOB, 10-90
providing download link, 10-89
BLOBs
APEX_UTIL.GET_BLOB_FILE_SRC, 10-91
download support, 11-22
populating, 11-22
support, 10-89
support in forms, 11-22
working with procedurally, 10-91
Bookmark Hash Function attribute, 7-24
branch
about, 17-25
creating, 16-20, 16-21
page submission, 8-18
pre-rendering, 8-17
When Button Pressed attribute, 2-39
BRANCH_TO_PAGE_ACCEPT, 2-40
branching
making conditional, 16-22
on load, before header, 16-22
on submit, after processing, 16-22
on submit, before computation, 16-22
on submit, before processing, 16-22
on submit, before validation, 16-22
using buttons, 15-49
breadcrumb reports, 16-15
Breadcrumb Exceptions, 16-15
Breadcrumb Hierarchy, 16-15
Breadcrumb History, 16-15
Breadcrumb Utilization, 16-15
Grid Edit Breadcrumb, 16-15
breadcrumb template, 13-33
Definition, 13-34
Name, 13-34
Substitution Strings, 13-35
breadcrumb template substitution strings
#COMPONENT_CSS_CLASSES#, 13-35
#LINK#, 13-35
#LONG_NAME#, 13-35
#NAME#, 13-35
NAME_ESC_SC#, 13-35
breadcrumbs
about, 16-7
about creating manually, 16-9
adding a breadcrumb region, 16-11, 16-12
adding entries, 16-10
adding to a page, 16-11, 16-12
creating, 16-7
creating as shared components, 16-9
creating dynamic, 16-13
creating from Component View, 16-9
creating from Page Designer, 16-9
deleting unused, 16-14
defining, 16-13
Exceptions report, 16-15
Grid Edit report, 16-15
Hierarchy report, 16-15
History report, 16-15
navigations, 13-33
reparenting entries, 16-14
Utilization report, 16-15
Browse button, A-6
bug reports
Bugs Dashboard, 5-15
By Day, 5-16
By Day page, 5-16
By Developer, 5-15
By Developer page, 5-16
Calendar page, 5-15
bugs
attaching files, 5-14
Bugs page, 5-13
creating, 5-13
defining, 5-14
managing, 5-13
viewing, 5-13
viewing reports, 5-15
build options, 24-31
changing status, 24-32
creating, 24-31
excluding, 24-32
exporting, 24-32
exporting status, 24-32
including, 24-32
reports, 24-33
Build Options attribute, 7-19
Build Status attribute, 7-17
built-in substitution strings, 2-26
Bulk Edit Item Help report, 12-43
button
  branching, 15-49
  creating, 15-42, 15-43, 15-44
  creating multiple at once, 15-46
  delete multiple, 15-47
  displaying conditionally, 15-43, 15-48
  editing, 15-43, 15-46
  editing multiple, 15-47
  managing in Component View, 15-44
  managing in Page Designer, 15-42
  names, 15-49
  template, 13-35
button template
  Definition, 13-36
  Name, 13-35
  Substitution Strings, 13-36
button template substitution strings
  ##ICON_CSS_CLASSES##, 13-36
  #BUTTON_ATTRIBUTES#, 13-36, 13-37
  #BUTTON_CSS_CLASSES#, 13-36
  #BUTTON_ID#, 13-36
  #LABEL#, 13-36, 13-37
  #LINK#, 13-36, 13-37

C
Cache attribute, 7-24
caching regions, 14-7, 14-18
Calendar
  adding to new page, 12-3
  accessing attributes, 12-7, 12-10
  adding Calendar, 12-3
  adding Legacy Calendar, 12-5
  creating, 12-1
  creating a Calendar, 12-9
  editing attributes, 12-7, 12-10
  enabling drag and drop, 12-12
  including add and edit functions, 12-8, 12-11
  managing in Component View, 12-9
  supported substitution strings, 12-12
  viewing sample, 12-2
calendar attributes
  accessing, 12-7, 12-10
cascading style sheets
  about, 13-24
  referencing for SVG charts, 12-36
  referencing in page template, 13-59
  referencing inline (SVG charts), 12-36
  uploading, 13-59
  using custom, 13-58
Central pane, 8-2
Change History report, 7-63
Character Value Comparison, 7-28
Character Value Comparison Behavior, 7-28
chart
  creating with Create Application Wizard, 7-2
  chart synchronous update
    creating refresh dynamic action, 12-29, 12-32
  editing chart attributes, 12-28, 12-31
  chart syntax, 12-23
  charts
    about Flash, 12-23
    about wizards, 12-23
    adding to a page, 12-26, 12-30
    adding to new page, 12-25
    altering display, 12-27, 12-31
    asynchronous updates, 12-28, 12-31
    candlestick syntax, 12-24
    chart syntax, 12-23
    creating, 12-22
    dial chart syntax, 12-24
    displaying SVG charts in other languages, 12-37
    editing attributes, 12-27, 12-31
    enabling asynchronous updates, 12-28, 12-31
    Gantt syntax, 12-24
    HTML chart support, 12-37
    HTML5 chart about, 12-22
    managing in Component View, 12-29
    managing in Page Designer, 12-26
    migrating SVG to Flash, 12-33
    monitoring information, 12-28, 12-31
    multiple series syntax, 12-24
    providing a SQL query, 12-23
    range syntax, 12-24
    scatter syntax, 12-24
    SQL query syntax, 12-23
    SVG chart support, 12-32
    switching to Flash, 12-28, 12-31
    types, 12-23
    upgrading a SVG chart to HTML5, 12-33
    upgrading all SVG charts to HTML5, 12-33
    upgrading SVG charts to HTML5, 12-33
  check box, A-3
    creating dependent LOV, 15-20
  classic calendar template
    comments, 13-37
    Daily Calendar, 13-38
    Monthly Calendar, 13-38
    Name, 13-37
    Template Subscription, 13-37
    Weekly Calendar, 13-38
  classic report
    adding download link, 10-68, 10-78
    adding with Create Page Wizard, 7-38, 10-5
    changing, 10-64, 10-72
    changing alignment, 10-64, 10-72
    changing column attributes, 10-64, 10-72
    changing column text, 10-64, 10-72
    changing pagination, 10-67, 10-75
    controlling column breaks, 10-72, 10-82
    controlling column display, 10-71, 10-82
    creating, 10-3
    creating a sum, 10-64, 10-72
    creating updatable columns, 10-70, 10-80
    defining a column as a list of values, 10-71, 10-81
editing attributes, 10-65, 10-73
editing report layout, 10-66, 10-74
exporting, 10-69, 10-78
hiding columns, 10-64, 10-72
sorting columns, 10-64, 10-72
synchronizing column order with region source, 10-83, 11-12
classic report regions
configuring print attributes, 10-85
printing methods, 10-84
classic reports
about, 10-2
Cleanup PL/SQL Code, 7-25
collections, 25-6
color picker, A-3
columns templates, using JavaScript, 13-58
Comments attribute, 9-21
Compatibility Mode attribute, 7-16
component
attaching an authorization scheme, 20-53
Component Settings
editing, 18-7
managing, 18-7
viewing History report, 18-8
Component View, 9-5
adding a page control, 9-10
Branches, 9-9
Breadcrumbs, 9-10
Buttons, 9-8
Create, Page Group, 9-4
Computation, 9-9
configuring dialog attributes, 9-23
Copy icon, 9-6
copying page, 9-26
Create icon, 9-6
Create menu, 9-4
Create, Developer Comment, 9-4
Create, Page, 9-4
Create, Page as Copy, 9-4
Create, Page Component, 9-4
Create, Region, 9-4
Create, Shared Components, 9-4
Create, Team Development, 9-4
deleting application page, 9-26
Edit All Icon, 9-6
Items, 9-8
List of Values, 9-10
Lists, 9-10
marking region titles as translatable, 21-4
Navigation Bar, 9-10
Next Page button, 9-4
Page, 9-8
Page Designer button, 9-5
Page Finder, 9-4
Page Processing, 9-8
Page Rendering, 9-7
Previous Page button, 9-4
Processes, 9-9
Regions, 9-8
Runtime Developer toolbar, 9-25
Security, 9-10
Tabs, 9-10
Templates, 9-10
Theme, 9-10
using, 9-5
Utilities, 9-4
Utilities, Application Utilities, 9-5
Utilities, Attribute Dictionary, 9-4
Utilities, Caching, 9-4
Utilities, Clean up, 9-4
Utilities, Cross Page Utilities, 9-5
Utilities, Delete, 9-4
Utilities, Delete Page, 9-4
Utilities, Export, 9-5
Utilities, History, 9-5
Utilities, Lock, 9-4
Utilities, Page Events, 9-4
Utilities, Page Groups, 9-5
Utilities, Resequence Components, 9-4
viewing field-level help, 1-21
Component view
Edit All Icon, 9-6
running a page, 9-24
using, 9-6
Component View toolbar
about, 9-3
Create, Team Development, 9-3
Edit Page, 9-3
Shared Components, 9-3
Component View, page attributes, 9-11
Authentication, 9-17, 9-18
Authorization Scheme, 9-17
Browser Cache, 9-18
Cache Page, 9-20
Cache Page Condition, 9-20
Cache Timeout, 9-20
Comments, 9-21
Configuration, 9-20
CSS, Inline, 9-15
Cursor Focus, 9-13
Deep Linking, 9-17
Dialog, Attributes, 9-15
Dialog, CSS Classes, 9-16
Dialog, Height, 9-15
Dialog, Maximum Width, 9-15
Dialog, Width, 9-15
display attributes, 9-12
Duplicate Submission, 9-19
Error Handling, 9-20
Error Handling Function, 9-21
Expression 1, 9-20
File URLs, 9-14
Footer, 9-16
Form Auto Complete, 9-18
Groups, 9-12
Header and Footer, 9-16
Header Text, 9-16
Help, 9-21
In-Line Error Notification Text, 9-21
JavaScript, Execute when Page Loads, 9-14
Index-5
JavaScript, File URLs, 9-14
JavaScript, Function and Global Variable Declaration, 9-14
Name, 9-12
Page Alias, 9-12
Page CSS Classes, 9-13
Page Mode, 9-13
Page Template Options, 9-13
Read Only, 9-16
Rejoin Sessions, 9-19
security, 9-17
Server Cache, 9-20
Title, 9-13
Component View, Page Processing
Branches, 9-9
Computations, 9-9
Processes, 9-9
Component View, Page Rendering
Buttons, 9-8
Dynamic Actions, 9-8
Items, 9-8
Page, 9-8
Processes, 9-8
Regions, 9-8
Component View, Shared Components
Breadcrumbs, 9-10
List of Values, 9-10
Lists, 9-10
Navigation Bar, 9-10
Security, 9-10
Tabs, 9-10
Templates, 9-10
Theme, 9-10
components
accessing with keyboard, E-8
controlling access to, 2-14
displaying on all pages, 14-3
displaying on every page, 14-1
translating messages, 21-13, 21-14
components, managing availability, 20-10
condition type, 2-13
conditional
processing, 2-12
rendering, 2-12
conditions
displaying regions, 14-17
list of, C-1
using, 2-12
using with buttons, 15-43, 15-48
configuration, controlling, 24-31
confirmation page
about creating, 11-3, 14-12
controls, controlling access to, 2-14
Create Application Wizard, 2-1
about, 7-2
adding a column toggle report, 7-2, 10-4
adding blank page, 7-2
adding chart, 7-2
adding form, 7-3
adding list view and form, 7-3, 10-4
adding list view report, 7-3, 10-4
adding master detail, 7-3
adding reflow report, 7-3, 10-4
adding report, 7-3, 10-4
adding report and form, 7-3, 10-4
adding tabular form, 7-3
creating forms, 11-5
creating reports, 10-8
page options, 7-2
report options, 10-4
Create Branch Wizard, 16-20, 16-21
Create Navigation Bar Entry Wizard, 12-44, 16-16, 16-18, 16-19
Create Page Computation Wizard, 17-1, 17-4, 25-20
Create Page Process Wizard, 17-21
Create Page Wizard, 7-36
adding a calendar, 7-37
adding a chart, 7-37
adding a column toggle report, 7-39, 10-5
adding a login page, 7-38
adding a map, 7-38
adding a Report on Web service, 7-39, 10-6
adding a tree, 7-39
adding blank page, 7-37
adding classic report, 7-38, 10-5
adding interactive report, 7-39, 10-5
adding List View, 7-39, 10-6
adding plug-ins, 7-38
adding reflow report, 7-39, 10-6
breadcrumb, 14-13
calendar, 14-13
chart, 14-13
creating a feedback page, 7-37
creating a page with access control list, 7-37
creating a wizard, 7-39
creating Data Load Wizard
Data Load Wizard, 25-1
creating data loading wizard, 7-37
creating forms, 11-4
creating reports, 10-9
Form and Report on Web Service, 7-37, 11-3
form and report on Web service, 14-13
Form on a Procedure, 7-37, 11-2
Form on a SQL Query, 7-37, 11-2
form on a SQL query, 14-12
Form on a Table with Report, 7-37, 11-2
Form on Table or View, 7-37, 11-2
Form on Table with List View, 7-37, 11-2
Form on Web Service, 7-38, 11-3
form on Web service, 14-13
form options, 11-2
list, 14-13
map, 14-13
Master Detail Form, 7-38, 11-2
plug-ins, 14-13
report options, 10-5
Summary Page, 7-38, 11-3
summary page, 14-12
Tabular Form, 7-38, 11-2
tree, 14-13
Create Region Wizard,  12-9, 12-20, 12-30, 14-14, 14-22, 14-23, 16-12, 18-32
adding a column toggle report, 10-7
adding column toggle report, 14-11
adding interactive report, 10-7
adding reflow report, 10-7, 14-12
classic report, 14-11
creating forms, 11-6
Form and Report on Web Service, 11-4
Form on a Procedure, 11-3, 14-12
Form on a SQL Query, 11-4
Form on a Table with Report, 11-3, 14-12
Form on Table or View, 11-3, 14-12
Form on Table with List View, 11-3, 14-12
Form on Web Service, 11-4
form options, 11-3
Help text, 14-14
interactive report, 14-11
list view, 14-11
Master Detail Form, 11-4
master detail form, 14-12
Multiple Static Content, 14-11
PL/SQL dynamic content, 14-13
region display selector, 14-14
Report on Web service result, 14-11
report options, 10-6
Static Content, 14-11
Summary Page, 11-4
Tabular Form, 11-4
tabular form, 14-12
URL, 14-14
Create Report Region Wizard
accessing from Page Designer, 10-9
Create Validations Wizard, 17-9, 17-12, 17-13, 17-14
cross page utilities, 7-66
accessing, 7-66
Grid Edit of all Pages, 7-67
Page Attributes, 7-67
cross site-scripting
about, 20-16
about safe item types, 20-17
dynamic output, 20-18
escaping values, 20-17
HTML region, 20-17
protecting dynamic output, 20-18
protecting form items, 20-19
protecting HTML regions, 20-17
protecting report regions, 20-18
protecting static areas, 20-17
report regions, 20-18
restricting characters on items, 20-20
static display areas, 20-17
understanding session state protection, 20-21
using safe item types, 20-18
CSS classes, SVG charts, 12-34
CSS Finder, 7-45
Custom Authentication
setting up, 20-41
custom packaged application
about, 24-6
about Supporting Objects page, 24-7
adding access control list, 24-9
build options, 24-8
creating supporting objects, 24-7
defining an upgrade script, 24-11
deinstallation scripts, 24-9
deinstalling, 24-12
deleting installation options, 24-10
deleting messages, 24-10
deleting supporting object scripts, 24-10
how to create, 24-6
installation scripts, 24-8
installing supporting objects, 24-10
messages, 24-8, 24-9
prerequisites, 24-8
substitutions strings, 24-8
testing installation scripts, 24-10
upgrade messages, 24-9
upgrade scripts, 24-9
upgrading, 24-11
validations, 24-8
viewing an install summary, 24-12
D
DAD Credentials Verification
setting up, 20-45
Data Load Wizard
about creating, 25-1
about pages created, 25-2
adding to application, 25-1
creating, 25-2
creating with Create Page Wizard, 25-1
Data Load Definition, 25-2
re-creating pages, 25-4
data loading
adding, 25-1
Database Account Credentials authentication
setting up, 20-42
database application
about, 2-2
adding a page, 7-36
adding components, 7-36
adding pages, 7-36
assigning to groups, 7-60
based on queries, 7-4
based spreadsheet, 7-7
copying, 7-8
copying from Application home page, 7-9
copying with Create Application Wizard, 7-8
creating groups, 7-60
creating on tables, 7-4
deleting, 7-13
groups, 7-59
removing from groups, 7-60
running, 2-9
database application groups
assigning pages, 7-60
creating, 7-60
removing, 7-60
Index-8

removing pages, 7-60
database application page, 9-26
  adding, 7-36
  assigning to groups, 7-51
  copying in Component View, 9-26
  copying in Page Designer, 8-25
  creating from Application home, 7-40
  deleting, 8-26
  deleting a range of, 9-27
  deleting in Component View, 9-26
  deleting in Page Designer, 8-26
  deleting multiple, 8-26, 9-27
  deleting range of, 8-27
determining lock status, 7-53
grouping, 7-50
locking, 7-52, 7-53
managing, 7-36
preventing conflicts, 7-52
removing from groups, 7-52
running, 2-9
unlocking, 7-52, 7-54
database application page groups, 7-51
  assigning pages, 7-51
  creating, 7-51
  editing group definition, 7-51
  removing, 7-52
  removing pages, 7-52
  viewing, 7-50
database link, 25-26
  about, 25-26
  creating, 25-26
  viewing existing database links, 25-26
Database Object Dependencies report, 7-64
Database Session attribute, 7-25
database users
  granted privileges, D-1
  granting privileges, D-1
date picker, A-4
date picker (classic), A-4
DBMS_APPLICATION_INFO, 22-3
debugging, 22-1, 23-1
  checking for errors, 22-1
  enabling and disabling, 22-4
  isolating a problem, 22-7
  running with application, 22-5
  SQL queries, 23-2
  SQL tracing, 22-3
  using debug mode, 22-3
  using f?p to access, 22-5
  viewing reports at runtime, 22-6
  viewing reports from Component View, 22-6
  viewing reports in development mode, 22-5
Debugging attribute, 7-16
deep linking, 20-38
Deep Linking attribute, 7-22
Define Authorization Scheme, 7-21
defining, 25-9
deleting export files, 24-29
demonstration application
  about Sample Database Application, 3-6
deployment, 24-5
  creating supporting objects, 24-7
  exporting a page, 24-18
  exporting a Websheet, 24-19
  exporting application definition, 24-14
  exporting feedback, 24-22
  exporting script files, 24-22
  exporting static files, 24-19
  exporting themes, 24-19, 24-20
  exporting User Interface Defaults, 24-20
  exporting workspaces, 24-17
  importing application from SQL*Plus, 24-29
  importing export files, 24-23
  importing files, 24-23
  installing application export, 24-28
  installing files, 24-29
  managing database objects, 24-5
  of an application, 24-3
  options, 24-4
  publishing the URL, 24-30
  publishing the Websheet application URL, 24-30
  understanding the process, 24-3
developer comments
  adding in Page Designer, 7-56
  adding to an application, 7-56
  adding to application, 7-56
  deleting, 7-57
  editing, 7-56
  viewing, 7-56
  viewing reports, 7-58
Developer Comments calendar, 7-58
Developer Navigation tools
  about, 1-13
  Account menu, 1-16
  Administration menu, 1-14
  global search, 1-13
  Help menu, 1-15
Developer toolbar
  Application, 3-15, 8-24, 9-25
  Debug, 3-15, 8-25, 9-25
  Developer Toolbar Options, 3-15, 8-25, 9-25
  Edit Page, 3-15, 8-25, 9-25
  Home, 3-15, 8-24, 9-25
  Quick Edit, 3-15, 8-25, 9-25
  Session, 2-15, 3-15, 8-25, 9-25
  Show Grid, 3-15, 8-25, 9-25
  Theme Roller, 3-15, 8-25, 9-25
  View Debug, 3-15, 8-25, 9-25
Developer Toolbar Options
  Auto Hide, 3-15, 8-25, 9-25
  Display Position, 3-16, 8-25, 9-25
  Show Icons Only, 3-15, 8-25, 9-25
development process
  managing, 5-1
development process, managing, 5-1
dial chart syntax, 12-24
dialog
  configuring attributes, 9-23
  configuring attributes in Page Designer, 8-22
dialog page template, 7-41
dialog page template attributes, 13-48
dialog pages, 7-40
  about supported, 7-40
  branching, 7-42
  creating a dialog page template, 7-41
  defining, 7-41, 7-42
  generating a Cancel dynamic action, 7-42
  generating a close dialog page process, 7-42
  handling refresh actions, 7-43
  Modal Dialog, 7-40
  non-modal dialog, 7-40
  required template attributes, 13-48
  using dialog page template, 7-41
DML
  about locking, 25-27
  APEX_DML_LOCK_WAIT_TIME, 25-27
  FSP_DML_LOCK_ROW, 25-27
documentation
  about, 1-19
  accessing, 1-19
  searching downloaded library, 1-21
documents
  downloading, 1-20
  Downloading, 11-22
Dynamic Actions
  adding Actions, 15-39
  creating, 15-34, 15-37
  debugging, 15-41
  deleting, 15-37
  editing, 15-35, 15-39
  frequency and scope, 15-36, 15-40
  managing, 15-30
  managing in Component View, 15-37
  managing in Page Designer, 15-34
  understanding, 15-30
dynamic list
  about creating, 18-28
  creating from scratch, 18-30
  syntax, 18-28
dynamic translation, 21-34

E
Edit All icon, 9-6
Edit Application Definition
  Allow Feedback, 7-16
  Application Alias, 7-15
  Application Groups, 7-15
  Build Status, 7-17
  Compatibility Mode, 7-16
  Debugging, 7-16
  Email from Address, 7-16
  Error Handling, Default Error Display
    Location, 7-18
  Error Handling, Error Handling Function, 7-18
  Global Notifications, 7-18
  Logging, 7-15
  Message for unavailable application, 7-17
  Name attribute, 7-15
  Proxy Server, 7-16
  restrict access, 7-17
  Status, 7-17
  Version, 7-15
Edit Application Definition page
  about, 7-14
  accessing, 7-14
Edit Application page
  Database Session, 7-25
Edit Globalization Attributes, 7-26
  Application Date Format, 7-27
  Application Date Time Format, 7-27
  Application Language Derived From, 7-27
  Application Primary Language, 7-27
  Application Timestamp Format, 7-28
  Application Timestamp Time Zone Format, 7-28
  Automatic CSV Encoding, 7-29
  Automatic Time Zone, 7-28
  Character Value Comparison, 7-28
  Character Value Comparison Behavior, 7-28
Edit Security Attributes, 7-19
  accessing, 7-19
  Authentication, 7-20
  Authorization, 7-21
  Session Management, 7-21
  Session State Protection, 7-23
email
  configuring for instance, 19-2
  email log, 19-1
  email queue, 19-1
  enabling, 20-8
  managing, 19-1
  problems sending, 20-10
  sending with APEX_MAIL package, 19-2
Email from Address attribute, 7-16
Embed in Frames attribute, 7-25
embedded PL/SQL gateway
  security practices, 20-3
engine, 1-2
environment, configuring, 1-3
error message
  creating for a page process, 17-20, 17-24
  creating for application processes, 17-28
  errors, handling, 7-17, 9-20
Expire Bookmarks attribute, 7-24
export
  application, 24-14
  application attributes, 24-16
  build option status, 24-16
  database application, 24-12
  database application components, 24-12
  feedback, 24-22
  managing database objects, 24-5
  page, 24-18
  page components, 24-16
  plug-in, 24-20
  script file, 24-22
  shared components, 24-16
  static files, 24-19
  text for translations, 21-7
  themes, 24-19, 24-20
User Interface Defaults, 24-20
Websheet, 24-19
workspaces, 24-17
Export Repository, 24-28, 24-29
accessing, 24-28
exported application, importing, 24-23
Export/Install icon, 3-16, 4-5

f?p syntax, 2-17
feature reports
Calendar page, 5-6
Change History, 5-6
Feature Utilities, 5-8
Focus Areas, 5-7
Owners, 5-7
Progress Log, 5-7
Tree page, 5-6
feature utilities
assigning milestones, 5-21
change milestones for overdue features, 5-21
set feature due date to milestone, 5-21
Feature utilities
push due date of open features, 5-21
features
attaching files, 5-4
creating, 5-4
tracking, 5-3
updating, 5-4
viewing, 5-3
viewing reports, 5-5
feedback
adding a page, 5-16
Feedback page, 5-16
reviewing comments, 5-18
using, 5-17
viewing, 5-16
viewing reports, 5-18
feedback reports
Calendar, 5-18
Tag Cloud, 5-3
field-level Help, 1-21
viewing in Component View, 1-21
viewing in Page Designer, 1-22
file upload table
security risks, 20-32
Find icon, 7-45, 9-3
Flash chart
adding to new page, 12-25
candlestick syntax, 12-24
Gantt syntax, 12-24
multiple series syntax, 12-24
range syntax, 12-24
saving locally, 12-38
scatter syntax, 12-24
using custom XML, 12-38
focus areas
editing, 5-21
footer, substitution strings, 14-6, 14-17
form
creating with Create App Wizard, 7-3
forms
accessing using JAWS, E-5
Automatic Row Processing (DML) process, 11-18
creating a master detail, 11-12
creating a tabular form, 11-6
creating manually, 11-18
creating using Create Page Wizard, 11-4
creating with Create Application Wizard, 11-5
creating with Create Region Wizard, 11-6
populating, 11-20
understanding processing, 11-18
validating input, 11-21
full development environment, 1-2
functions, searching for, 7-49

Gallery
about, 8-9
using, 8-15, 8-16, 8-17
Global Notifications attribute, 7-18
Global page, 14-3
creating, 14-3
creating navigation bar entry, 14-3
navigating to, 14-3
global template option group, 13-27
creating, 13-27
editing, 13-27
global template options
creating, 13-29
editing, 13-29
managing, 13-27
Global Templates, 13-12
global variable, creating, 18-17
globalization
attributes, 21-4
determining application language, 7-26
format masks, 21-5
understanding, 21-1
graphics
managing, 14-25
groups
of applications, 7-59
of database application pages, 7-50

Help
accessing, 1-19
creating, 12-39
creating navigation bar entry, 12-44
editing, 12-43
searching, 1-19
using, 1-19
Help menu, 1-15
Help text
accessing with JAWS, E-6
defining for item, 12-42
defining for page, 12-42
saving alternative, 10-63
saving as private, 10-61
saving as public, 10-61
saving primary default, 10-63
user save options, 10-60
using, 10-57
using filters, 10-59
valid linking examples, 10-31, 10-54
viewing as a developer, 10-56
viewing saved report activity, 10-56
internal messages, requiring translation, 21-15
IP address, restricting access, 20-4
isolating workspaces, about, 20-6
item
  naming conventions, 15-2
  referencing values, 15-2
  searching for, 7-46
  using format masks when translating, 21-5
Item Finder, 7-45, 7-46
item Help
  editing, 12-43
  viewing, 1-21
item type
  Check box, A-3
  Color Picker, A-3
  Date Picker, A-4
  Date Picker (classic), A-4
  Display Image, A-5
  Display Only, A-5
  File Browse, A-6
  Hidden, A-6
  List Manager, A-7
  Number Field, A-7
  Password, A-8
  Popup List of Values, A-9
  Radio, A-10
  Rich Text Editor, A-11
  Select list, A-12
  Shuttle, A-13
  slider, A-14
  Text Area, A-17

J
JavaScript, 13-47
  adding using Component View, 9-21
  adding using Page Designer, 8-20
  calling from button, 15-48
  in row templates, 13-58
  including in .js file, 13-24
  including in page template, 13-24
  incorporating, 15-57
  libraries, 13-47
  referencing items, 15-3
  referencing jQuery library, 19-13
  setting focus on item, 9-13
  using a dynamic action, 15-41
  using in a plug-in, 19-14
  using jQuery, 19-13
JavaScript Finder, 7-45

JAWS, E-1
  about validation errors, E-6
  accessing forms, E-5
  accessing Help text, E-6
jQuery
  support, 19-10
  using in JavaScript, 19-13
jQuery Mobile, 7-10
  support, 7-10
jQuery UI
  accessing other jQuery UI components, 19-12
  available features, 19-12
  support, 19-10
jsTree, support, 15-50

K
keyboard shortcuts, E-11

L
label template
  #CURRENT_ITEM_HELP_TEXT#, 13-39
  creating page item Help, 13-39
  Definition, 13-38
  editing, 13-38
  Error Display, 13-39
  Field Container, 13-39
  Name, 13-38
  Substitution Strings, 13-39
language
  defining for application, 21-4
  multibyte, 21-6
  preference, 21-2
layout
  about region attributes, 14-15
  controlling, 13-1
LDAP Directory Verification
  setting up, 20-43
Legacy Calendar
  adding to new page, 12-5
legacy calendar template, 13-37
limits
  classic reports, B-1
  forms, B-1
  interactive reports, B-1
  item names, B-2
  tabular forms, B-1
  validations, B-2
Link Column
  about, 10-20, 10-43
  excluding, 10-22, 10-45
  linking a single row view, 10-20, 10-44
  linking to a page, 10-21, 10-44
  linking to URL, 10-22, 10-45
  linking
    deep, 20-38
list
  about creating, 18-20, 18-21
  about dynamic, 18-28
Index-13

about static, 18-22
adding to a page, 18-32
Conditional Entries report, 18-33
copying, 18-30
copying between applications, 18-31
copying from current application, 18-31
creating hierarchical, 13-40
ing, 18-32
ing attributes, 18-32
History report, 18-34
Lists page, 18-32
reports, 18-33
types, 18-21
Unused report, 18-33
Utilization report, 18-33
list manager, creating dependent LOV, 15-20
list of values, 15-20
bulk edit of static LOVs, 18-19
Conditional Entries report, 18-20
creating, 18-18
creating at application-level, 18-18
Edit Null Text report, 18-20
ing, 18-19
Grid Edit report, 18-20
History report, 18-20
Locally Defined report, 18-20
named, 18-18
referencing session state, 18-20
reports, 18-20
shared, 18-18
static, 18-19
Subscription report, 18-20
translating, 21-34
Utilization report, 18-20
list reports
  Conditional Entries, 18-33
  History, 18-33
  Unused, 18-33
  Utilization, 18-33
list template
  After List Entry, 13-42
  After Sub List Entry, 13-41
  Before List Entry, 13-40
  Before Sub List Entry, 13-41
ing, 13-40
  Name, 13-40
  Sub List Entry, 13-41
  Substitution Strings, 13-42, 13-45
  Template Definition, 13-40
  Template Definitions for First Entry, 13-41
List View
  adding with Create Page Wizard, 7-39, 10-6
  adding with Create Region Wizard, 14-11
list view and form
  creating with Create Application Wizard, 7-3, 10-4
list view report
  creating with Create Application Wizard, 7-3, 10-4
lists of terms, A-7
Lists page, 18-32
Lists Wizard, 18-23, 18-28
locks
determining lock status, 7-53
for database application page, 7-52
Logging attribute, 7-15
login controls, creating, 20-4
login credentials
  recovering workspace name, 1-9
login page
  creating, 20-48
logo
defining, 7-31
Logo attribute, 7-31
logs
  managing, 23-3
  performance, 23-3
M
Manage Session State Protection attribute, 7-24
maps
  about map chart support, 12-13
  about wizards, 12-13
  adding to a page, 12-17, 12-20
  adding to new page, 12-16
  creating, 12-12
  creating SQL queries, 12-13
  editing map attributes, 12-18, 12-21
  enabling asynchronous updates, 12-22
  enabling automatic updates (Page Designer), 12-19
  managing in Component View, 12-19
  managing in Page Designer, 12-17
  providing a SQL query, 12-13
  using custom XML, 12-19, 12-21
master application, creating, 13-18
master detail
  creating with Create Application Wizard, 7-3
  master detail form, creating, 11-12
Maximum Session Idle Time in Second attribute, 7-23
Maximum Session Length in Seconds attribute, 7-22
menu, 1-13, 1-16
Message for unavailable application, 7-17
messages
  creating for translation, 21-14
  editing text messages, 21-15
  in PL/SQL procedures, 21-13
  in reports
    translating, 21-14
  translating, 21-13
  translating internal, 21-14
messages in reports, 21-14
messages requiring translation
  interactive reports, 21-17
  internal, 21-15
milestone reports
  By Owner report, 5-9
  Feature by Milestone report, 5-10
Milestone Calendar, 5-9
milestones
creating, 5-8
tracking, 5-8
updating, 5-9
viewing, 5-8
viewing reports, 5-9
mobile applications, 7-10
about designing, 2-8
jQuery Mobile, 7-10
mobile design, 2-9
responsive design, 2-9
Mobile Theme - 51, 13-2
monitor activity
Calendar Reports, 23-4
Developer Activity, 23-5
Environment Reports, 23-4
Login Attempts, 23-5
Logs, 23-5
Page Views, 23-4
Service Requests, 23-4
Workspace Purge, 23-4
multiple select list item, 15-4
handling return values, 15-4
using APEX_UTIL.STRING_TO_TABLE, 15-4

N
navigation
branch, 16-20
breadcrumb, 16-7
list, 18-20
parent tabs, 16-1
standard tabs, 16-1
tab, 16-1
navigation bar, 16-15
navigation bar entry, 16-15
about, 16-16
copying, 16-18
creating, 16-16
creating for feedback, 16-18
creating from scratch, 16-16
editing, 16-19
editing multiple, 16-19
History report, 16-20
reports, 16-20
Subscription report, 16-20
navigation bar entry reports
history, 16-20
subscription, 16-20
news entries, 1-12
adding, 5-22
deleting, 5-23
editing, 5-23
number field, A-7
NV function, 2-21

O
online Help, 1-19

Packaged Applications
about, 3-2
accessing, 3-2, 3-3
Packaged Apps Only workspace, 3-5
accessing, 3-2, 3-3
Packaged App Administration, 3-28
Packaged App Gallery, 3-27, 3-28

Oracle Application Express
about, 1-1
architecture, 1-2
browser requirements, 1-3
component limits, B-1
configuring your environment, 1-3
keyboard shortcuts, E-11
signing in, 1-2
signing in to as a developer, 1-6
user roles, 1-3, 20-2
Oracle Application Express administrator, 1-3, 20-2
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, 1-2
Oracle BI Publisher, integrating with, 20-10
Oracle Help Center
searching, 1-19, 1-20
Oracle HTTP Server
about security, 20-2
Oracle XML DB Protocol Server, verifying HTTP server port, 1-9
output escaping
substitution strings, 2-26, 13-25

problems, 20-10
searching, 1-19
Open Door Credentials
setting up, 20-46
Oracle Application Express
about, 1-1
architecture, 1-2
browser requirements, 1-3
component limits, B-1
configuring your environment, 1-3
keyboard shortcuts, E-11
signing in, 1-2
signing in to as a developer, 1-6
user roles, 1-3, 20-2
Oracle Application Express administrator, 1-3, 20-2
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, 1-2
Oracle BI Publisher, integrating with, 20-10
Oracle Help Center
searching, 1-19, 1-20
Oracle HTTP Server
about security, 20-2
Oracle XML DB Protocol Server, verifying HTTP server port, 1-9
output escaping
substitution strings, 2-26, 13-25

problems, 20-10
searching, 1-19
Open Door Credentials
setting up, 20-46
Oracle Application Express
about, 1-1
architecture, 1-2
browser requirements, 1-3
component limits, B-1
configuring your environment, 1-3
keyboard shortcuts, E-11
signing in, 1-2
signing in to as a developer, 1-6
user roles, 1-3, 20-2
Oracle Application Express administrator, 1-3, 20-2
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, 1-2
Oracle BI Publisher, integrating with, 20-10
Oracle Help Center
searching, 1-19, 1-20
Oracle HTTP Server
about security, 20-2
Oracle XML DB Protocol Server, verifying HTTP server port, 1-9
output escaping
substitution strings, 2-26, 13-25

problems, 20-10
searching, 1-19
Open Door Credentials
setting up, 20-46
Oracle Application Express
about, 1-1
architecture, 1-2
browser requirements, 1-3
component limits, B-1
configuring your environment, 1-3
keyboard shortcuts, E-11
signing in, 1-2
signing in to as a developer, 1-6
user roles, 1-3, 20-2
Oracle Application Express administrator, 1-3, 20-2
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, 1-2
Oracle BI Publisher, integrating with, 20-10
Oracle Help Center
searching, 1-19, 1-20
Oracle HTTP Server
about security, 20-2
Oracle XML DB Protocol Server, verifying HTTP server port, 1-9
output escaping
substitution strings, 2-26, 13-25

problems, 20-10
searching, 1-19
Open Door Credentials
setting up, 20-46
Oracle Application Express
about, 1-1
architecture, 1-2
browser requirements, 1-3
component limits, B-1
configuring your environment, 1-3
keyboard shortcuts, E-11
signing in, 1-2
signing in to as a developer, 1-6
user roles, 1-3, 20-2
Oracle Application Express administrator, 1-3, 20-2
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, 1-2
Oracle BI Publisher, integrating with, 20-10
Oracle Help Center
searching, 1-19, 1-20
Oracle HTTP Server
about security, 20-2
Oracle XML DB Protocol Server, verifying HTTP server port, 1-9
output escaping
substitution strings, 2-26, 13-25
All Page Items, 15-16
Conditional Items, 15-16
Grid Edit of all Item Help Text, 15-16
Grid Edit of all Item Labels, 15-16
Item Help Subscriptions, 15-16
page-level template, 13-25

pages
about, 2-4
about Global page, 14-3
calling from a button, 2-20
calling with an alias, 2-19
controlling access to, 20-36
duplicate submission, 9-19
enabling user customization, 14-6, 14-18
exporting, 24-18
linking with ?p syntax, 2-17
optimizing for printing, 14-2
resource use, 22-3
searching for, 7-47
server cache, 9-20
specifying page template, 13-25

Pages Finder, 7-47

pagination
classic report, 10-67, 10-75
interactive reports, 10-13, 10-36
reports, 13-52, 13-57

Pagination Subtemplate
Next Page Template, 13-58
Next Set Template, 13-58
Pagination Template, 13-58
Previous Page Template, 13-58
Previous Set Template, 13-58

parent tabs, creating, 16-1
Parsing Schema, 7-25
password
best practices, 20-14
changing, 1-18
identifying risks, 20-15
resetting from Sing In page, 1-9
restricting reuse, 20-6
Security Profiles Report, 20-15

passwords
Password Items Report, 20-16
patch, creating, 24-16
PDF printing, problems, 20-10

performance, 23-1
best practices, 23-1
database parameters, 23-5
debugging problematic SQL queries, 23-2
diagnosing issues, 23-2
identifying issues, 23-1
limiting resources, 23-5
slow queries, 23-2
slow running applications, 23-2
uploading static files to Web server, 23-6
using logs and report, 23-3
PL/SQL code, searching for, 7-49
PL/SQL Finder, 7-45
plug-ins
about, 19-3
accessing, 19-4
adding events, 19-8
creating, 19-4
custom attributes, 19-6
deleting, 19-8
duplicate submission, 20-36
editing, 19-6
examples, 19-3
exporting, 19-9
implementing, 19-3
importing, 19-8, 24-25
resetting, 19-9
uploading associated files, 19-7
using JavaScript code, 19-14
utilization, 19-9
viewing history, 19-10
viewing Plug-in Repository, 19-8

pop up list of values, A-9

Popup LOV template
Buttons, 13-49
ingredients, 13-48
Page Attributes, 13-49
Pagination, 13-49
Result Set, 13-49
Search Field, 13-49
Window, 13-49

preconfigured authentication schemes, 20-39
Printer Friendly template, 14-2
printing reports, 10-83
privileges
about granting to database users, D-1
generated to database users, D-1
generated to PUBLIC, D-1

Procedure Finder, 7-49

procedures, searching for, 7-49

process
Cache Management, 17-18, 17-21, 17-22
Data Manipulation, 17-18, 17-21, 17-22
Form Pagination, 17-18, 17-21, 17-22
On Demand, 17-18, 17-21, 17-22
PL/SQL, 17-18, 17-21, 17-22
Report Pagination, 17-18, 17-21, 17-22
Web Services, 17-18, 17-21, 17-22

productivity application, unlocking, 3-5
programming techniques
collections, 25-6
database links, 25-26
implementing Web services, 25-7

Property Editor, 8-2
proxy server, 25-9

Proxy Server attribute, 7-16
PUBLIC
privileges granted to, D-1
public file upload, enabling, 20-4
public synonyms, D-1
Public User attribute, 7-20

Q
Query Finder, 7-45, 7-47
query, searching for, 7-47

R

radio group, A-10
LOV, 15-20
Recently Updated Pages report, 7-63
reflow reports
creating with Create Application Wizard, 7-3, 10-4, 14-12
creating with Create Page Wizard, 7-39, 10-6
creating with Region Page Wizard, 10-7
region
about types, 14-10
accessing utilities and reports, 14-21
adding an image, 14-8, 14-20
based on a URL, 14-24
controlling positioning, 14-6, 14-17
copying, 14-9, 14-23
creating, 14-4, 14-14
creating for a list, 18-32
creating region display selector, 14-9, 14-22
creating subregion, 14-21
deleting, 14-10, 14-23
deleting multiple regions at once, 14-24
displaying conditionally, 14-17
editing attributes, 14-5, 14-15, 14-22
editing attributes for multiple regions, 14-20
editing region attributes, 14-5, 14-15, 14-22
enabling customization, 14-6, 14-18
enabling users to customize a page, 14-6, 14-18
how attributes affect layout, 14-15
specifying a static region ID, 14-8, 14-20
specifying header and footer, 14-6, 14-17
understanding, 14-4, 14-10
region attributes
Cache, 14-7, 14-18
Column, 14-16
Column Attributes, 14-16
Column CSS Classes, 14-16
Column Span, 14-16
Condition Type, 14-17
Customization, 14-17
Display Point, 14-15
Expression, 14-17
Image Tag Attributes, 14-16
Item Display Position, 14-15
Read Only Condition Type, 14-17
Region CSS Classes, 14-16
Region Display Selector, 14-16
Region Footer, 14-16
Region Header, 14-16
Region Image, 14-16
Sequence, 14-15
Start New Grid, 14-16
Static ID, 14-8, 14-20
Template, 14-15
region display selector, creating, 14-9, 14-22
region template
Definition, 13-50
editing, 13-49
Form Table Attributes, 13-51
Name, 13-50
Sub Regions Attributes, 13-51
Substitution Strings, 13-51
region titles
marking as translatable, 21-3
region type
Breadcrumb, 14-13
Calendar, 14-13
Chart, 14-13
Classic Report, 14-11
Form and Report on Web Service, 14-13
Form on a Procedure, 14-12
Form on a SQL Query, 14-12
Form on a Table or View, 14-12
Form on a Table with List View, 14-12
Form on a Table with Report, 14-12
Form on Web Service, 14-13
Help Text, 14-14
Interactive Report, 14-11
List, 14-13
List View, 14-11
Map, 14-13
master detail form, 14-12
Multiple Static Content, 14-11
PL/SQL Dynamic Content, 14-13
Plug-ins, 14-13
Region Display Selector, 14-14
Report on Web Service Result, 14-11
Static Content, 14-11
Summary Page, 14-12
Tabular Form, 14-12
Tree, 14-13
URL, 14-14
Rejoin Sessions
about security risks, 20-5
configuring at application-level, 7-22
configuring at page-level, 8-21, 9-22
Rejoin Sessions attribute, 7-22
report
creating with Create Application Wizard, 7-3, 10-4
Oracle database, 23-5
report and form
creating with Create Application Wizard, 7-3, 10-4
report column templates
colorizing cell headings, 13-53
omitting headings, 13-53
report column templates (generic)
After Rows, 13-54
Before Each Row, 13-53, 13-54
Before Rows, 13-52
Column Headings, 13-53
Column Templates, 13-53
displaying HTML after all columns, 13-54
displaying HTML after last row, 13-54
Next Page Template, 13-55
Next Set Template, 13-55
Pagination Subtemplate, 13-55
Previous Page Template, 13-55
Previous Set Template, 13-55
Report Template, 13-52
Row Highlighting, 13-55
report column templates (named)
  Before Rows attribute, 13-57
  colorizing column header cell, 13-57
  Column Headings, 13-57
Pagination Subtemplate, 13-57
Report Template Identification, 13-56
Row Templates, 13-56
Subscription, 13-56
report layouts
  about, 18-41
  copying, 18-43
  creating, 18-42
  editing, 18-43
  options, 18-41
Report on Web service
  creating with Create Page Wizard, 7-39, 10-6
Report on Web service result
  creating with Create Region Wizard, 14-11
report printing, problems, 20-10
report queries
  about, 18-38
  copying, 18-40
  creating, 18-38
  editing, 18-40
report regions
  about configuration options, 10-84
  printing, 10-83
  printing to PDF, 10-84
report templates, 13-51
  editing, 13-51
  generic column templates, 13-52
  named column templates, 13-52
  using JavaScript, 13-58
reports, 13-29
  about, 10-1
  about database application, 7-61
Active Session History (ASH) reports, 23-5
adding by dragging and dropping, 10-9
Application Dashboard, 7-62, 23-3
Automatic Workload Repository (AWR)
  reports, 23-5
Bulk Edit Item Help, 12-43
Change History, 7-63
class references, 13-23
creating a column link, 10-69, 10-79
creating with Create Application Wizard, 10-8
creating with Create Page Wizard, 10-9
creating without SQL knowledge, 10-9
CSV export, 13-53
database applications, 7-61
extporting to CSV format, 13-52, 13-57
including pagination above, 13-52, 13-57
monitor activity, 23-4
performance, 23-3
printing, 10-83
Recently Updated Pages, 7-63
sorting columns, 10-68, 10-77
supported substitution strings, 13-23
template subscription, 13-29
template utilization, 13-29
templates in a theme, 13-21
theme file references, 13-22
theme template counts, 13-22
themes, 13-21
understanding differences, 10-1
workspace activity, 23-3, 23-4
REQUEST
  about scope and value, 2-39
  about When Button Pressed attribute, 2-39
  button names, 15-49
  referencing using declarative conditions, 2-40
  referencing value of, 2-38
  substitution string, 2-38
  using for Show processing, 2-40
resource use, monitoring, 22-3
responsive design, 2-9
responsive design, about, 13-2
RESTful access, enabling, 20-5
restrict access, to application, 7-17
Run Application button, 4-5
Run Application icon, 3-16
Run on Public Pages attribute, 7-21
Runtime API Usage, 7-26
Runtime Developer toolbar, 8-24, 9-25
runtime environment, 1-2
  advantages, 20-10
S
safe item types, 20-17
Sample Database Application
  about, 3-6
  Administration page, 3-12
  Customers page, 3-9
  editing, 3-14, 3-16
  editing an application, 3-16
  Home page, 3-8
  installing, 3-6
  Mobile page, 3-13
  modifying, 3-14, 3-16
  running, 3-6
  understanding, 3-6, 3-7
screen reader mode, E-1
  about, E-2, E-3
  controlling with APIs, E-2
  controlling with f?p syntax, E-2
  controlling with substitution string, E-2
  enabling, E-1
  extending, E-3
  provisioning, E-2
SDLC
  rapid application development, 24-3
  Spiral, 24-3
Search Workspace, 1-13
secure sockets layer, best practices, 20-5
security attributes, 7-19
for pages, 9-17

security attributes
- Allows URLs Created After, 7-24
- authentication, 7-20
- Authentication Scheme, 7-20
- Authorization Scheme, 7-21
- Bookmark Hash Function, 7-24
- Browser Security, 7-24
- Cache, 7-24
- Database Session, 7-25, 7-26
- Deep Linking, 7-22
- Define Authorization Schemes, 7-21
- Embed in Frames, 7-25
- Expire Bookmarks, 7-24
- HTML Escaping Mode, 7-25
- HTTP Response Headers, 7-25
- Manage Session State Protection, 7-24
- Maximum Session Idle Time in Seconds, 7-23
- Maximum Session Length in Seconds, 7-22
- Public User, 7-20
- Rejoin Sessions, 7-22
- Run on Public Pages, 7-21
- Session Idle Timeout URL, 7-23
- Session Management, 7-21
- Session State Protection, 7-23
- Session Timeout URL, 7-23

security best practices
- configuring instance settings, 20-8
- configuring storage options, 20-8
- configuring workspace settings, 20-9
- controlling SQL access in Websheets, 20-7
- creating login controls, 20-4
- creating wallets, 20-9
- cross site-scripting protection, 20-16
- embedded PL/SQL gateway, 20-3
- enabling email, 20-8
- enabling network services, 20-10
- enabling public file upload, 20-4
- enabling RESTful access, 20-5
- enabling RESTful services, 20-8
- enabling self service workspace provisioning, 20-8
- for administrators, 20-1
- for developers, 20-14
- identifying risk passwords, 20-15
- isolating workspaces, 20-6
- items of type password, 20-14
- managing component availability, 20-10
- managing instance security, 20-3
- managing users, 20-9
- managing workspace to schema assignments, 20-9
- managing workspaces, 20-9
- Oracle BI Publisher, 20-10
- Oracle HTTP Server, 20-2
- restricting access by IP address, 20-4
- restricting password reuse, 20-6
- runtime environment, 20-10
- secure sockets layer, 20-5
- session joins, 20-5
- session state, 20-20
- session timeout, 20-6
- specifying instance proxy, 20-4
- security risks, file upload table, 20-32
- seeding, 21-7
- select list, A-12
- select lists, creating dependent LOV, 15-20
- session ID, 2-15
- Session Idle Timeout URL, 7-23
- session joins
- preventing browser attacks, 20-5
- security best practices, 20-5
- Session Management, 7-21
- specifying, 7-21
- session state
- clearing, 2-22
- clearing application cache, 2-23
- 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, 2-22
- enabling projection, 7-23
- management, 2-14
- passing item value, 2-23
- protection, 20-21
- referencing, 2-20
- resetting an application completely, 2-23
- security best practices, 20-23
- setting, 2-21
- sharing across applications, 18-17
- viewing, 2-15
- session state protection
- configuring all attributes, 20-24
- configuring security attributes, 20-23
- disabling, 20-21
- enabling, 20-21
- enabling from Edit Security Attributes, 20-21
- enabling from Session State Protection, 20-22
- enabling using a wizard, 20-22
- reports, 20-23
- understanding, 20-21
- viewing existing settings, 20-23
- viewing summaries, 20-23
- Session State Protection attribute, 7-23
- session state values, managing, 2-20
- Session Timeout URL, 7-23
- session timeout, utilizing, 20-6
- shared components
- about, 9-9, 18-2
- accessing, 18-2
- backing up, 24-16
- creating, 18-2
- exporting, 24-16
- icon, 3-16, 4-5
- Shared Components page, 18-1
- Shared Components page
  Application Computations, 18-3
  Application Definition Attributes, 18-3
Application Items, 18-3
Application Processes, 17-25, 18-3
Authentication Schemes, 18-3
Authorization Schemes, 18-4
Breadcrumbs, 18-5
Build Options, 18-4
Component Settings, 18-4
Data Loading, 18-6
Developer Comments Calendar, 18-7
Edit Application Comments, 18-7
Export Application Components, 18-7
Globalization Attributes, 18-7
List of Values, 18-4
Lists, 18-5
Manage Supporting Objects, 18-7
Navigation Bar Entries, 18-5
Navigation Menu, 18-5
Plug-ins, 18-4
Report Layouts, 18-6
Report Queries, 18-6
Review Developer Comments, 18-7
Security Attributes, 18-3
Session State Protection, 18-4
Shortcuts, 18-4
Static Application Files, 18-6
Static Workspace Files, 18-6
Tabs, 18-5
Tasks Region, 18-7
Templates, 18-5
Text Messages, 18-7
Themes, 18-5
Translate Application, 18-7
User Interface Attributes, 18-5
Web Service References, 18-6
shared interactive reports, linking to, 10-32, 10-55
shortcuts, 18-34
about, 18-34
copying an existing shortcut, 18-36
defining, 18-35, 18-36
defining from scratch, 18-35
editing attributes, 18-37
History report, 18-37
HTML Text, 18-34, 18-36
HTML Text with Escaped Special
Characters, 18-35, 18-36
Image, 18-35, 18-36
Message, 18-35, 18-36
Message with JavaScript Escaped Special
Quotes, 18-35, 18-36
PL/SQL Function Body, 18-34, 18-36
referencing, 18-36
reports, 18-37
Subscription report, 18-37
supporting translatable messages, 21-3
Text with JavaScript Escaped Single
Quotes, 18-35, 18-36
types, 18-34
Show Page, 2-12
shuttle multiple select list, A-13
shuttle, creating dependent LOV, 15-20
sign in credentials, 1-7
sign in page, 1-2
understanding login credentials, 1-7
signing in to, Oracle Application Express, 1-6
signing out, 1-9
Single Sign-On
setting up, 20-46
slider item type, A-14
SOAP, 25-7
SQL, 6-10
SQL tracing, enabling, 22-3
standard tabs, creating, 16-1
static application files
deleting, 18-11
deleting all, 18-11
downloading, 18-10
downloading in Zip file, 18-11
managing, 18-8
referencing using #APP_IMAGES#, 18-10
uploading, 18-9
viewing details, 18-10
static files
exporting, 24-19
importing into another instance, 24-19
static list
about, 18-22
adding list entries, 18-24
adding sublists, 18-24
copying entries, 18-26
creating from scratch, 18-23
managing orphaned list entries, 18-27
reparenting entries, 18-27
resequencing entries, 18-27
static substitution string, 7-19
static workspace files
deleting, 18-14
deleting all, 18-14
downloading, 18-13
downloading all in Zip file, 18-14
managing, 18-12
referencing using #WORKSPACE_
IMAGES#, 18-13
uploading, 18-12
viewing details, 18-13
Status attribute, 7-17
storage options, configuring, 20-8
style sheet, 13-24
subregion, 14-21
creating, 14-21
creating by editing attributes, 14-22
using, 14-21
substitution output escaping
ATTR, 2-26
HTML, 2-26
JS, 2-26
preserve, do not escape, 2-26
RAW, 2-26
remove HTML tag, 2-26
reserved characters in HTML attribute, 2-26
reserved characters in JavaScript context, 2-26
SVG chart
   displaying in other languages, 12-37
   migrating to Flash, 12-33
   referencing CSS styles inline, 12-36
   referencing custom CSS, 12-36
   upgrade restrictions, 12-33
   upgrading all to HTML5, 12-33
   upgrading to HTML5, 12-33
SVG chart, about CSS classes, 12-34
SVG syntax, for dynamic list, 18-28
system message, 1-12

T

tab
   about Tabs page, 16-2
   adding, 16-1
   Conditional Display report, 16-7
   create Standard tab, 16-6
   create Standard tab set, 16-6
   creating, 16-2, 16-4
   editing, 16-5
   editing multiple, 16-6
   History report, 16-7
   managing, 16-2
   move standard to different parent, 16-6
   Parent, 16-2
   rename standard, 16-6
   reports, 16-6
   resequence, 16-6
   Standard, 16-2
   Standard Tab Task, 16-6
   template support, 16-3
   Utilization report, 16-7

tab reports
   Conditional Display, 16-7
   History, 16-7
   Table Finder, 7-45
   Tables Finder, 7-48
   tables, searching for, 7-48
   Tabs page, 16-2

tabular form
   creating, 11-6
   creating a page process, 17-22
   creating with Create Application Wizard, 7-3
Tag Cloud, 5-3
Team Development
   accessing, 5-1
   Features dashboard, 5-3, 5-6
   home page, 5-1
   managing bugs, 5-13
   managing feedback, 5-16, 5-19
   managing links, 5-23
   managing news entries, 5-22
   managing To Dos, 5-10
   News Region, 5-2
   purging data, 5-22
   tracking features, 5-3
   tracking milestones, 5-8
   Utilities, 5-2

Index-22
Team Development file attachments
viewing, 5-22
Team Development utilities, 5-19
accessing, 5-19
assigning milestones, 5-21
change milestones for overdue features, 5-21
configuring Team Development settings, 5-20
editing focus areas, 5-21
enable support for file upload, 5-20
manage focus areas, 5-20
manage links, 5-20
manage milestones for multiple features, 5-20
manage news, 5-20
manage uploaded files, 5-20
managing links, 5-23
managing news entries, 5-22
Purge Data, 5-20
purging data, 5-22
push due date of open features, 5-21
push past due bugs, 5-20
Release Summary, 5-21
Release Summary report, 5-20
set feature due date to milestone, 5-21
Team Development settings, 5-19
update assignee, 5-20
viewing file attachments, 5-22
template edit history, 13-29
template option presets, 13-7
template options
about, 13-6
default, 13-7
groups, 13-7
presets, 13-7
using, 13-6
template options groups, 13-7
templates
breadcrumbs, 13-33
button, 13-35
changing defaults, 13-8
columns, 13-53, 13-58
creating, 13-26
customizing, 13-23
labels, 13-38
legacy calendar, 13-37
lists, 13-40
managing, 13-30
page, 13-42
popup LOV, 13-48
publishing, 13-33
regions, 13-49
replacing, 13-31
replacing all, 13-32
reports, 13-51
selecting for a page, 13-25
selecting theme default, 13-25
supported substitution strings, 13-23
unsubscribing to, 13-32
viewing for specific page, 13-31
viewing History report, 13-30
viewing on Templates page, 13-30
viewing Publish report, 13-30
viewing region position utilization, 13-32
viewing reports, 13-29
viewing Subscription report, 13-30
viewing Utilization report, 13-30
text area, A-17
text strings, 21-14
translating, 21-13
text strings, translating, 21-13
tHEME attributes, 13-8
Breadcrumb, 13-10, 13-11
Button, 13-10
Charts, 13-11
Classic Report, 13-10
CSS File URLs, 13-9
Custom Classes, 13-12
Custom Image, 13-12
Custom Library File URLs, 13-12
Custom Prefix Class, 13-12
Date Picker Icon Attributes, 13-12
Date Picker Icon Name, 13-12
Default Label, 13-10
dialog button region, 13-11
dialog content region, 13-11
dialog page, 13-11
dialog page, 13-11
dialog page, 13-11
error page, 13-10
Files, 13-13
Footer Toolbar, 13-11
forms, 13-11
global template options, 13-12
header toolbar, 13-11
interactive reports, 13-11
JavaScript File URLs, 13-11
Legacy Calendar, 13-10
Library, 13-12
List, 13-10
Lists, 13-11
Login Page, 13-10
Name, Application, 13-8
Name, Description, 13-9
Name, Name, 13-8
Name, Navigation Bar Implementation, 13-9
Name, Navigation Type, 13-9
Name, Theme Number, 13-8
Name, User Interface, 13-9
Navigation Bar List, 13-10
Navigation Menu List (Top), 13-10
Navigation Menu List Position, 13-10
Optional Label, 13-10
Page, 13-9
Page Template Options, 13-10
Page Transition, 13-11
popup transition, 13-12
popup transition, 13-12
popup transition, 13-12
printer friendly page, 13-10
Region, 13-10
Report, 13-11
Required Label, 13-10
styles, 13-12
Tabular Forms, 13-11
Theme Subscription, 13-9
Wizards, 13-11
theme reports
Application Templates, 13-21
Class References, 13-23
class references, 13-23
File References, 13-22
files references, 13-22
supported substitution strings, 13-23
template counts, 13-22
Template Substitution Strings, 13-23
templates in a theme, 13-21
Theme Template Counts, 13-22
Theme Roller, 3-15, 8-25, 9-25
using, 13-4
theme styles, 13-3
about, 13-3
creating, 13-3
deleting, 13-4
editing, 13-4
Theme Roller, 13-4
using, 13-1
theme subscriptions
about, 13-13
modifying, 13-14
refreshing, 13-14
re-subscribing, 13-15
unsubscribing, 13-15
verifying refresh, 13-14
themes
about, 13-1
about subscriptions, 13-13
accessing, 13-7
changing ID, 13-17
changing identification number (ID), 13-17
copying, 13-18
creating, 13-15
creating master application, 13-18
default templates, 13-8
deleting, 13-18
editing, 13-8
editing theme attributes, 13-8
exporting, 13-21, 24-19, 24-20
exporting and importing, 13-21
exporting with different ID, 13-17
grid-based, 13-2
importing, 13-21, 24-26
managing, 13-1
reports, 13-21
selecting page-level template, 13-25
switching, 13-16
switching to Universal Theme, 13-16
theme styles, 13-3
Themes page
accessing from Shared Components, 13-7
To Do reports
Calendar page, 5-12
Progress Log, 5-12
To Dos
attaching files, 5-11
creating, 5-11
managing, 5-10
To Dos page, 5-10
updating a task, 5-11
viewing, 5-10
viewing reports, 5-12
translatable messages, defining, 21-13
translating, 21-14
messages in PL/SQL procedures, 21-13
translation, 21-1
applications, 21-1
dynamic, 21-34
dynamic text strings, 21-3
dynamic translations, 21-3
editing manually, 21-12
exporting text, 21-7
globalization support, 21-1
language identification, 21-2
mapping primary application ID, 21-6
mapping target application ID, 21-6
messages, 21-3
region titles, 21-3
rules, 21-2
seeding, 21-7
shortcuts, 21-3
steps, 21-6
templates, 21-4
translation file, 21-8
understanding, 21-6
understanding application rendering, 21-2
XLIFF, 21-8
XLIFF Target Elements, 21-9
translation file, 21-8
publishing, 21-8
uploading, 21-10
Tree pane, 8-2
trees
about, 15-50
creating, 15-49, 15-52, 15-55
creating on new page, 15-50
editing attributes, 15-54, 15-56
managing in Component View, 15-54
managing in Page Designer, 15-52
U
UDDI registry, 25-9
Universal Theme
Migration Guide, 13-16
switching to, 13-16
Universal Theme - 42, 13-2
unlocking, productivity application, 3-5
Upgrade Application icon, 7-63, 7-64
URL
publishing, 24-30
SSL-enabled, 14-25
syntax, 2-16
user identity
establishing, 20-37
verifying, 2-13
user interface
about, 1-1
about mobile applications, 2-9
about selecting, 2-8
controlling, 13-1
managing design of, 2-8
User Interface attributes
Add New User Interface, 7-31
configuring, 7-29
CSS file URLs, 7-31
Image Prefix, 7-31
Logo, 7-31
Media Type, 7-31
Static File Prefix, 7-30
User Interface Detection, 7-31
User Interfaces, 7-31
User Interface Defaults
exporting, 24-20
importing, 24-26
User Interface Details, 7-32
Auto Detect, 7-33
Concatenated Files, 7-36
Content Delivery Network, 7-34
Default, 7-33
Display Name, 7-33
File URLs, 7-35
Global Page, 7-33
Home URL, 7-33
Include jQuery Migrate, 7-35
Include Legacy JavaScript, 7-35
List Template, 7-34
Login URL, 7-33
Navigation List, 7-34
Sequence, 7-33
Template Options, 7-34
Theme, 7-33
Theme Style, 7-33
User Interface Type, 7-33
user roles
developer, 1-3, 20-2
Oracle Application Express administrator, 1-3, 20-2
Workspace administrator, 1-3, 20-2
users
about managing, 20-9
utilities, 7-61
accessing for application, 7-61

V
V function, 2-21
validation errors, running JAWS, E-6
validations
about, 17-6, 17-7
about Go to error link, E-7
associating with item, 11-21
creating, 17-7, 17-8
creating for item, 17-7
creating for page, 17-7, 17-11
creating for page item, 17-9
creating for tabular form column, 17-13
creating for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11
defining for page item, 17-9
defining for tabular form column, 17-13
defining for tabular form row, 17-14
defining with error messages, 11-21
defining error messages, 17-16
defining for item, 17-7
defining for page, 17-7, 17-11

W
WAI-ARIA support, E-8
wallets, creating, 20-9
Web service process
editing a process, 25-22
mapping input parameters to static values, 25-22
Web service reference, 25-8
creating, 25-9
creating forms, 25-17
creating forms and reports, 25-15
creating manually, 25-11, 25-12
specifying WSDL, 25-10
viewing a history, 25-23
Web Service Reference page, 25-8
Web services, 25-7
creating a reference from a WSDL, 25-10
creating RESTful Web services, 25-12
displaying Web service results in a report, 25-21
editing a Web service process, 25-22
invoking as a process, 25-20
problems, 20-10
RESTful Web service
exposing a report region, 25-23
specifying proxy server address, 25-9
SSL enabled, 25-9
testing a RESTful Web service reference, 25-14
testing Web service reference manually, 25-12
using, 25-8
viewing Web service reference history, 25-23
Web service reference repository, 25-14
Web Services References page, 25-8
Websheet applications
about, 2-2, 6-1
access control list, 6-10
controlling access, 6-10
copying a page, 6-3
creating, 6-1
creating access control list entries, 6-11
creating from Application Builder, 6-2
creating from running Websheet, 6-3
deleting, 6-5
editing page details, 6-5
monitoring activity, 6-6
running, 6-3
specifying access control list, 6-11
understanding Websheet properties, 6-6
viewing Websheet dashboard, 6-5
viewing data within, 7-65
W
Websheet authentication
  Application Express Account, 6-9
  Custom, 6-9
  LDAP, 6-9
  Single Sign-On, 6-9
Websheet dashboard, viewing, 6-5
Websheet properties
  about, 6-7
  Access Control List Type, 6-9
  Allow Public Access, 6-9
  Application Date Format, 6-7
  Authentication, 6-9
  Cookie Domain, 6-9
  Cookie Name, 6-9
  Cookie Path, 6-9
  Default Application Language, 6-7
  Default Application Territory, 6-7
  Description, 6-8
  editing, 6-7
  editing SQL access, 6-10
  Home Page, 6-7
  Login Message, 6-8
  Logo, 6-8
  Logo Type, 6-8
  Logout URL, 6-9
  Name, 6-7
  Owner, 6-8
  Secure, 6-9
  Show Reset Password, 6-8
  Status, 6-7
  Style, 6-8
  Websheet Email From Address, 6-8
Websheets
  about navigation, 3-18
  editing, 3-24
  modifying, 3-24
  navigating, 3-17
  understanding navigation, 3-19
  using breadcrumbs, 3-18
Wizards
  Create Page, 7-36
  creating a branch, 16-21
  creating a calendar, 12-1
  creating a chart, 12-22
  creating a list, 18-23, 18-28
  creating a LOV, 18-18
  creating a page computation, 17-1, 17-4, 25-20
  creating a page process, 17-21
  creating a region, 14-14
  creating an application, 2-2
  creating applications, 2-1
  creating navigation bar entry, 12-44, 16-16, 16-18, 16-19
  creating validations, 17-9, 17-12, 17-13, 17-14
Workspace
  recovering workspace name, 1-9
  requesting, 1-6
  searching, 1-13
  signing in, 1-7
  signing out, 1-9
Workspace Activity reports
  Application Errors, 23-4
  Developer Activity, 23-4
  Environment, 23-4
  Login Attempts, 23-4
  Page View Analysis, 23-4
  Page Views, 23-4
  Sessions, 23-4
  Workspace Schema Reports, 23-4
Workspace administrator, 1-3, 20-2
Workspace home page
  Accessibility Mode region, 1-17
  Administration menu, 1-14
  Available Updates region, 1-16
  News entries, 1-12
  system message, 1-12
  Top Applications region, 1-12
  Top Users region, 1-12
workspace name
  recovering, 1-9
workspace news
  managing, 5-22
workspace provisioning, enabling self service, 20-8
workspace settings, configuring, 20-9
workspace themes
  adding, 13-19
  deleting, 13-20
  exporting, 13-21
  managing, 13-19, 13-21
  modifying, 13-20
  viewing existing, 13-20
workspace to schema assignments, managing, 20-9
workspaces, managing, 20-9
WSDL document, 25-8

X

XLIFF, 21-8
  applying, 21-10
  deleting an uploaded document, 21-11
  Target Elements, 21-9
  uploading, 21-10
  uploading and publishing, 21-10

Z

Zip file
  downloading static application files, 18-11
  downloading static workspace files, 18-14