

Oracle® APEX

Accessibility Guide



Release 23.1

F74106-01

May 2023

ORACLE®

Oracle APEX Accessibility Guide, Release 23.1

F74106-01

Copyright © 2017, 2023, Oracle and/or its affiliates.

Primary Author: Caryn Dietrich

Contributing Authors: Terri Jennings, Anthony Rayner, Chaitanya Koratamaddi

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

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

Contents

Preface

Audience	vi
Documentation Accessibility	vi
Diversity and Inclusion	vi
Related Documents	vii
Conventions	vii

1 Changes in Release 23.1 for Oracle APEX Accessibility Guide

2 About This Accessibility Guide

2.1 What is Web Accessibility?	2-1
2.2 Why Accessibility is Important	2-2
2.3 About Building for Accessibility	2-2

3 Using Oracle APEX with Assistive Technologies and Keyboard

3.1 Recommended Configuration	3-1
3.1.1 Enabling Accessibility Tags in PDF Documents	3-2
3.2 Accessing Form Pages Using JAWS	3-2
3.2.1 About the Structure of Pages	3-2
3.2.2 Accessing Help Text	3-3
3.2.3 Dealing with Validation Errors	3-3
3.2.3.1 Reviewing the Error	3-3
3.2.3.2 Navigating to the Relevant Fields to Fix the Errors	3-3
3.2.3.3 About WAI-ARIA Support	3-4
3.3 Accessing Oracle APEX Using a Keyboard	3-4
3.3.1 About General Keyboard Shortcuts	3-4
3.3.1.1 General	3-5
3.3.2 About Development Environment Keyboard Shortcuts	3-5
3.3.2.1 General Development Environment	3-5
3.3.2.2 Page Designer	3-6

3.3.2.3	SQL Commands	3-7
3.3.2.4	Code Editor	3-7
3.3.3	About Component-Specific Keyboard Shortcuts	3-8
3.3.3.1	Date Picker	3-8
3.3.3.2	CSS Calendar	3-9
3.3.3.3	Chart Region	3-12
3.3.3.4	Map Region	3-13
3.3.3.5	Interactive Grid Region	3-13
3.3.3.6	Markdown Editor	3-15
3.3.3.7	Rich Text Editor	3-15

4 Developing Accessible Apps

4.1	Themes	4-1
4.1.1	About Accessibility Features in Universal Theme	4-1
4.1.2	About Accessible Theme Styles	4-2
4.1.3	About Theme Roller	4-2
4.2	Pages	4-3
4.2.1	Page Titles	4-4
4.2.2	Page Modes	4-4
4.2.3	Cursor Focus	4-5
4.2.4	Reload on Submit	4-5
4.3	Regions	4-5
4.3.1	Page Structure	4-6
4.3.1.1	Landmarks	4-6
4.3.1.2	Headings	4-10
4.3.2	Report Regions	4-14
4.3.3	Interactive Grid Regions	4-14
4.3.4	Interactive Report Regions	4-14
4.3.5	Classic Report Regions	4-15
4.3.6	JET Chart Regions	4-16
4.4	Form Pages and Items	4-17
4.4.1	Form Page Attributes	4-17
4.4.2	Select Lists and Radio Groups	4-18
4.4.3	Display Image Alternative Text	4-19
4.5	Buttons	4-19
4.6	Testing Apps for Accessibility	4-21
4.6.1	About Accessibility Testing	4-21
4.6.2	About Accessibility Checks in Advisor	4-21
4.6.3	Running Advisor on a Single Page	4-22

5 Open Bugs and Known Issues

6 Further Reading

Index

Preface

Oracle APEX Accessibility Guide offers an introduction to accessibility in Oracle APEX. The guide contains information both for developers who are building applications and also end-users using applications built with APEX.

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

Audience

Oracle APEX Accessibility Guide is intended for developers building Oracle APEX applications. [Using Oracle APEX with Assistive Technologies and Keyboard](#) is intended for both developers and end users of applications built with APEX.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For more information, see these Oracle resources:

- *Oracle APEX Release Notes*
- *Oracle APEX Installation Guide*
- *Oracle APEX App Builder User's Guide*
- *Oracle APEX End User's Guide*
- *Oracle APEX Administration Guide*
- *Oracle APEX SQL Workshop Guide*
- *Oracle APEX API Reference*

Conventions

The following text conventions are used in this document:

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

1

Changes in Release 23.1 for *Oracle APEX Accessibility Guide*

All content in *Oracle APEX Accessibility Guide* is updated to reflect release 23.1 functionality and user interface changes.

New Features and Updates

For new features and updates, see New Features and Changed Behavior in *Oracle APEX Release Notes*.

Deprecated and Desupported Features

See Deprecated Features and Desupported Features in *Oracle APEX Release Notes*.

About This Accessibility Guide

This guide consists of two distinct sections: general instructions for accessing Oracle APEX with assistive technologies and a keyboard, and guidelines for developing APEX applications that are accessible and highly usable to all users.

"Oracle is committed to creating accessible technologies and products that enhance the overall workplace environment and contribute to the productivity of our employees, our customers, and our customers' customers." —Safra Catz, Chief Executive Officer, Oracle

Accessibility Overview

Web accessibility means people with disabilities can effectively use and contribute to the web. To ensure people with disabilities can use APEX, we need to design and develop more accessible web applications and software.

Accessibility is not just a checklist, but an ongoing and planned effort. The aim of this guide is to help you start the planning process, and understand how to build accessible applications with APEX.

- [What is Web Accessibility?](#)
Web accessibility allows people with disabilities to interact with and contribute to the web.
- [Why Accessibility is Important](#)
Accessible products improve experiences for everyone.
- [About Building for Accessibility](#)
Oracle tracks conformance to Section 508, WCAG, and other standards and guidelines in the Accessibility Conformance Report.

2.1 What is Web Accessibility?

Web accessibility allows people with disabilities to interact with and contribute to the web.

This guide refers to the [Web Content Accessibility Guidelines 2.1 \(WCAG 2.1\)](#) to back up some of the guidelines with the relevant standard and provide further reading.

According to W3C, web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the web, and that they can contribute to the web. The disabilities that affect access to the web can be categorized as: visual, auditory, speech, cognitive, physical, and neurological. The goal of web accessibility is to provide equal access to users with disabilities. That is, developers need to focus on building more accessible web applications and products. Apart from people with disabilities, the elderly with increasing impairments, and people in a limiting situation (for example, slow internet connection, no audio etc.) can also benefit with accessible applications and products.

Principles of Accessibility

WCAG 2.1, which is created by W3C, defines the four principles of accessibility as:

1. **Perceivable:** Information and user interface components must be presentable to users in ways they can perceive. That is, users must be able to perceive the information presented.

2. **Operable:** User interface components and navigation must be operable. That is, users must be able to operate the interface.
3. **Understandable:** Users must be able to understand the information as well as the operation of the user interface.
4. **Robust:** Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

2.2 Why Accessibility is Important

Accessible products improve experiences for everyone.

Up to 16% of the world population is disabled, either through birth, aging, illness, or the result of an accident. Accessible products remove obstacles between a company and its existing or potential customers. Accessibility enables employers to recruit from a broader pool of talent. Overall, accessibility generally improves products for everyone.

2.3 About Building for Accessibility

Oracle tracks conformance to Section 508, WCAG, and other standards and guidelines in the Accessibility Conformance Report.

Building for accessibility means coding to standards, and not for a specific technology. Building to standards often leads to better, standardized HTML and cleaner code. The process of building accessible applications at Oracle involves a considerable amount of time and resources at all stages of the development cycle.

The Oracle Accessibility Program Office, reporting to the office of the Chief Corporate Architect, is responsible for defining the corporate standards for accessibility, and developing materials to train all employees so that they can successfully create products that meet those standards. To learn more about Oracle's Accessibility Program, see [Oracle Accessibility Information and Resources](#).

Oracle uses the Accessibility Conformance Report (ACR) to represent the degree of conformance to various accessibility standards and guidelines, including Section 508 and WCAG. Depending on when a product was developed and released, different standards may be listed.

3

Using Oracle APEX with Assistive Technologies and Keyboard

Users can access Oracle APEX using only a keyboard or a screen reader such as Freedom Scientific's screen reader JAWS.



Note:

JAWS with all the 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.

- [Recommended Configuration](#)
Learn about recommended software.
- [Accessing Form Pages Using JAWS](#)
Learn about navigating page structure, help text, and validation errors with JAWS.
- [Accessing Oracle APEX Using a Keyboard](#)
Learn about accessing Oracle APEX using a keyboard.



See Also:

[Oracle's Accessibility Program](#)

3.1 Recommended Configuration

Learn about recommended software.

Oracle recommends the screen reader JAWS with the most recent version of Google Chrome for the best user experience.

When reading PDF files exported from APEX (for example, data exports from our report-based region types), Oracle recommends using Adobe Acrobat Reader DC for the best user experience with a screen reader.

- [Enabling Accessibility Tags in PDF Documents](#)
Toggle accessibility tags for PDF documents with the `APEX_UTIL.SET_PREFERENCE` procedure.

3.1.1 Enabling Accessibility Tags in PDF Documents

Toggle accessibility tags for PDF documents with the `APEX_UTIL.SET_PREFERENCE` procedure.

Creating an accessible PDF enables the document to be read aloud by a screen reader and other text-to-speech tools and increases the PDF file size. Setting the `p_preference` parameter `APEX_PDF_ACCESSIBLE` to `Y` adds accessibility tags for native PDF printing. Alternately, setting `APEX_PDF_ACCESSIBLE` to `N` disables accessibility tags. This preference is set automatically when the **Include Accessibility Tags** switch is enabled in the download dialog of an interactive report or interactive grid.



See Also:

`SET_PREFERENCE` Procedure in *Oracle APEX API Reference*

3.2 Accessing Form Pages Using JAWS

Learn about navigating page structure, help text, and validation errors with JAWS.

Many of the pages in Oracle APEX 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](#)
Learn about different regions for form pages.
- [Accessing Help Text](#)
Learn about accessing field-level help with a keyboard shortcut.
- [Dealing with Validation Errors](#)
Learn about features to help JAWS users address validation errors.

3.2.1 About the Structure of Pages

Learn about different regions for form pages.

Many form pages are logically broken up into different regions. For example, the Edit Application Definition page has the following regions containing form fields for maintaining information about the current application:

- Application (app ID)
- Name
- Properties
- Availability
- Error Handling
- Global Notification
- Substitutions

- Build Options
- Report Printing
- Copyright Banner
- About
- Tasks

Oracle APEX uses Heading (Hn) tags throughout its development interface to define page structure. Please use standard screen reader heading navigation to navigate such pages.

3.2.2 Accessing Help Text

Learn about accessing field-level help with a keyboard shortcut.

Most items within the Oracle APEX 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 **ALT+F1**. Pressing this keyboard shortcut displays a popup window containing the help text.

3.2.3 Dealing with Validation Errors

Learn about features to help JAWS users address validation errors.

Oracle APEX provides many built-in features that help a JAWS user deal with validation errors on form pages.

- [Reviewing the Error](#)
Learn about navigating to the list of errors.
- [Navigating to the Relevant Fields to Fix the Errors](#)
Learn about navigating to the page item related to the error.
- [About WAI-ARIA Support](#)
Learn about how WAI-ARIA markup helps users effectively address item errors.

3.2.3.1 Reviewing the Error

Learn about navigating to the list of errors.

To navigate to the section of the page where the error messages displays, locate the H2 (Heading Level 2) tag with the text "n error has occurred" (where "n" is the number of errors). Immediately after this heading, there is an unordered list containing all the errors that have occurred on the page.

3.2.3.2 Navigating to the Relevant Fields to Fix the Errors

Learn about navigating to the page item related to the error.

For validations that are associated with a particular page item, Oracle APEX renders the actual error message as a link in the notification section. When a user activates the link, focus shifts to the associated item. This enables all users to easily navigate straight to the page item related to the error.

3.2.3.3 About WAI-ARIA Support

Learn about how WAI-ARIA markup helps users effectively address item errors.

Oracle APEX includes some Web Accessibility Initiative - Accessible Rich Internet Applications (WAI-ARIA) markup to help screen reader usability in dealing effectively with item errors. 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 announces "Job Edit, Invalid Entry, Job must have some value, Type of Text".

3.3 Accessing Oracle APEX Using a Keyboard

Learn about accessing Oracle APEX using a keyboard.



Note:

The Oracle APEX interface is intended for use with the US keyboard layout. Users of other keyboard layouts should refer to the US keyboard layout in conjunction with this documentation as there may be incompatibilities regarding number and special character keys, such as the ' (quote) key.

- [About General Keyboard Shortcuts](#)
Learn about keyboard shortcuts for Oracle APEX.
- [About Development Environment Keyboard Shortcuts](#)
Learn about keyboard shortcuts for the development environment.
- [About Component-Specific Keyboard Shortcuts](#)
Learn about keyboard shortcuts for specific components.

3.3.1 About General Keyboard Shortcuts

Learn about keyboard shortcuts for Oracle APEX.

You can use general keyboard shortcuts throughout Oracle APEX.

If a page incorporates specialized keyboard shortcuts, click the **Help** icon located in the top-right of the page and select **Shortcuts**, or press **Alt+Shift+F1** to display a list of valid keyboard shortcuts. Not all pages have specialized shortcuts.

- [General](#)
Learn about general keyboard shortcuts for Oracle APEX.

3.3.1.1 General

Learn about general keyboard shortcuts for Oracle APEX.

Action	Where	Keys
Display the list of valid keyboard shortcuts for the page	Any page with special keyboard shortcuts	Windows: Alt+Shift+F1 Mac: Option+Shift+F1
Open field help dialog	When focus is on any field with a (?) help icon	Windows: Alt+F1 Mac: Option+F1
Move focus back to field without closing dialog	When focus is on a field help dialog	Windows: Alt+F6 Mac: Option+F6
Close dialog	When focus is on any dialog	Escape
Select the previous/next tab	Any page with focus in region display selector tabs	Arrow keys

3.3.2 About Development Environment Keyboard Shortcuts

Learn about keyboard shortcuts for the development environment.

Development environment keyboard shortcuts are specific to the development environment.

If a page incorporates specialized keyboard shortcuts, click the **Help** icon located in the top-right of the page and select **Shortcuts**, or press **Alt+Shift+F1** to display a list of valid keyboard shortcuts. Not all pages have specialized shortcuts.

- [General Development Environment](#)
Learn about keyboard shortcuts for the general development environment.
- [Page Designer](#)
Learn about keyboard shortcuts for Page Designer.
- [SQL Commands](#)
Learn about keyboard shortcuts for SQL Commands.
- [Code Editor](#)
Learn about keyboard shortcuts for Code Editor.

3.3.2.1 General Development Environment

Learn about keyboard shortcuts for the general development environment.

Action	Where	Keys
Display Global Search	Any page	Ctrl+Quote
Move splitter	Focus on a splitter handle	Arrow keys
Expand or collapse splitter, if supported by splitter	Focus on a splitter handle	Enter
Move to next splitter	On pages that have a splitter	Ctrl+F6

Action	Where	Keys
Move to previous splitter	On pages that have a splitter	Ctrl+Shift+F6

3.3.2.2 Page Designer

Learn about keyboard shortcuts for Page Designer.

Action	Keys
Display from here, in Layout tab	Ctrl+/, D, H
Display from page, in Layout tab	Ctrl+/, D, T
Go to Dynamic Actions tab	Ctrl+/, A
Go to Gallery Buttons	Ctrl+/, G, B
Go to Gallery Items	Ctrl+/, G, I
Go to Gallery Regions	Ctrl+/, G, R
Go to Help	Windows: Alt+F1 Mac: Option+F1
Go to Layout tab	Ctrl+/, L
Go to Messages	Ctrl+/, M
Go to Page Shared Components	Ctrl+/, S
Go to Processing tab	Ctrl+/, P
Go to Property Editor Attributes	Ctrl+/, E, A
Go to Property Editor Main	Ctrl+/, E, M
Go to Property Editor Printing	Ctrl+/, E, P
Go to Rendering tab	Ctrl+/, R
List keyboard shortcuts	Windows: Alt+Shift+F1 Mac: Option+Shift+F1
Navigate to next page	Windows: Alt+Shift+Page Down Mac: Option+Shift+Page Down
Navigate to previous page	Windows: Alt+Shift+Page Up Mac: Option+Shift+Page Up
Go to Page Search tab	Ctrl+/, F
Redo	Ctrl+Y
Restore/Expand the Central Pane	Windows: Alt+F11 Mac: Option+F11
Save	Windows: Alt+F7 Mac: Option+F7
Save and Run Page	Windows: Alt+F8 Mac: Option+F8
Toggle Hide Empty Positions	Ctrl+/, D, E
Undo	Ctrl+Z
Open Advisor	Ctrl+/, U, A
Open context menu, with focus anywhere that has a context menu	Shift+F10 or Context menu key
Move selection, in Layout Gallery and Icon List wizards	Arrow keys
Move to previous group, in Property Editor with focus on a group heading	Up arrow

Action	Keys
Move to next group, in Property Editor with focus on a group heading	Down arrow
Move to first group, in Property Editor with focus on a group heading	Home
Move to last group, in Property Editor with focus on a group heading	End
Expand or collapse the group, in Property Editor with focus on a group heading	Left or Right arrows or Enter or Space
Select previous or next tab, with focus on any tab	Arrow keys

3.3.2.3 SQL Commands

Learn about keyboard shortcuts for SQL Commands.

Action	Keys
Run the current command	Ctrl+Enter

3.3.2.4 Code Editor

Learn about keyboard shortcuts for Code Editor.

The Code Editor Keyboard Shortcuts table is not a complete list of shortcuts. To see all the keyboard shortcuts, open the Code Editor and press F1.

Action	Keys
Show complete list of commands	F1
Undo	Windows: Ctrl+Z Mac: Cmd+Z
Redo	Windows: Ctrl+Shift+Z Mac: Cmd+Shift+Z
Find	Windows: Ctrl+F Mac: Cmd+F
Replace	Windows: Ctrl+H Mac: Option+Cmd+F
Trigger Suggest , when available	Ctrl+Space
Insert a tab character or spaces, depending on the indentation setting	Tab

 Note:

Can also be used to move focus to the next field, depending on the **Tab Key Moves Focus** setting

Toggle **Tab Key Moves Focus**

Windows: Ctrl+M
Mac: Ctrl+Shift+M

Action	Keys
Save , on a Global/Modal page	Windows: Alt+F7, Alt+F8 Mac: Option+F7, Option+F8
Save and Run Page , on a Standard page	Windows: Alt+F7, Alt+F8 Mac: Option+F7, Option+F8

3.3.3 About Component-Specific Keyboard Shortcuts

Learn about keyboard shortcuts for specific components.

Component-specific keyboard shortcuts only work for specific components.

If a page incorporates specialized keyboard shortcuts, click the **Help** icon located in the top-right of the page and select **Shortcuts**, or press **Alt+Shift+F1** to display a list of valid keyboard shortcuts. Not all pages have specialized shortcuts.

- [Date Picker](#)
Learn about keyboard shortcuts for the Date Picker.
- [CSS Calendar](#)
Learn about keyboard shortcuts for the CSS Calendar.
- [Chart Region](#)
Learn about keyboard shortcuts for the chart region.
- [Map Region](#)
Learn about keyboard shortcuts for the map region.
- [Interactive Grid Region](#)
Learn about keyboard shortcuts for the interactive grid region.
- [Markdown Editor](#)
Learn about keyboard shortcuts for the Markdown editor.
- [Rich Text Editor](#)
Learn about keyboard shortcuts for the rich text editor.

3.3.3.1 Date Picker

Learn about keyboard shortcuts for the Date Picker.

Action	Keys
Open the popup, when the input has focus and Display Mode is set to popup	Down arrow
Close the popup and focus the input, when Display Mode is set to popup	Escape
Focus the next element	Tab • Note: When the Display Mode is set to popup and the last element is focused, pressing Tab moves focus to the first element

Action	Keys
Focus the previous element	Shift+Tab Note: When the Display Mode is set to popup and the first element is focused, pressing Tab moves focus to the last element
Move focus to the same day of the previous week	Up arrow
Move focus to the same day of the next week	Down arrow
Move focus to the previous day	Left arrow
Move focus to the next day	Right arrow
Move focus to the first day of the current week	Home
Move focus to the last day of the current week	End
Change the grid of dates to the previous month	Page Up
Change the grid of dates to the next month	Page Down
Change the grid of dates to the previous year	Shift+Page Up
Change the grid of dates to the next year	Shift+Page Down
Select the focused date, when Show Time is off	Enter or Space
Select the focused date, when Show Time is on	Enter or Space, then Done

3.3.3.2 CSS Calendar

Learn about keyboard shortcuts for the CSS Calendar.

When you focus on the calendar grid (month, week or day view), you can use the arrow keys to navigate within the calendar. There are two calendar browsing modes:

1. **Event Browsing** - Focuses on existing calendar events
2. **Calendar Browsing** - Allows you to freely select dates or date ranges

The keyboard shortcut behavior depends on the current view.

Event Browsing CSS Calendar Keyboard Shortcuts

Action	Keys
Go to the previous event	Up arrow

 Note:

If on the first event in the given view, move to the last event.

Action	Keys
Go to the next event	Down arrow
 Note: If on the last event in the given view, move to the first one.	
Switch to Calendar Browsing mode	N
Leave browsing mode	Escape

Month View CSS Calendar Keyboard Shortcuts

Action	Keys
Go to the previous day	Left arrow
 Note: If on the first day of a month, switch to the previous month.	
Go to the next day	Right arrow
 Note: If on the last day of a month, switch to the next month.	
Go to the next week; same day	Down arrow
Go to the previous week; same day	Up arrow
Extend the selection by one day, up to the end of the current view	Shift+Right arrow
Decrease the selection by one day, only until the selection spans one day	Shift+Left arrow
Move to the next month	Page Down
Move to the previous month	Page Up
Run the Create Link given as Plug-in attribute by the developer or fire the Calendar Date Select event	Enter / Space
Leave Calendar Browsing and go back to Event Browsing	Escape

Week View CSS Calendar Keyboard Shortcuts

Action	Keys
Go to the previous day	Left arrow
 Note: If on the first day of a week, switch to the previous week.	
Go to the next day	Right arrow
 Note: If on the last day of a month, switch to the next month.	
Extend the selection by one day, up to the end of the current view	Shift+Right arrow
Decrease the selection by one day until the selection spans one day	Shift+Left arrow
Move selection 30 minutes back in time	Up arrow
Move selection 30 minutes forward in time	Down arrow
Extend selection range by 30 minutes	Shift+ Down arrow
Decrease selection by 30 minutes, until the selection is 30 minutes long	Shift+Up arrow
Move to the next week	Page Down
Move to the previous week	Page Up
Run the Create Link given as plug-in attribute by the developer or fire Calendar Date Select event	Enter / Space
Leave Calendar Browsing and go back to Event Browsing	Escape

Day View CSS Calendar Keyboard Shortcuts

Action	Keys
Go to the previous day	Left arrow
Go to the next day	Right arrow
Move selection 30 minutes back in time	Up arrow
Move selection 30 minutes forward in time	Down arrow
Extend selection range by 30 minutes	Shift+Down arrow
Decrease selection by 30 minutes, until the selection is 30 minutes long	Shift+Up arrow
Move to the next day	Page Down
Move to the previous day	Page Up

Action	Keys
Run the Create Link given as plug-in attribute by the developer or fire Calendar Date Select event	Enter / Space
Leave Calendar Browsing and go back to Event Browsing	Escape

3.3.3.3 Chart Region

Learn about keyboard shortcuts for the chart region.

Action	Keys
Move focus to next element	Tab
Move focus to previous element	Shift+Tab
Move focus and selection to previous data item	Up arrow
Move focus and selection to next data item	Down arrow
Move focus and selection to previous data item (on left)	Left arrow
Move focus and selection to next data item (on right)	Right arrow
Move focus and multiselect previous data item	Shift+Up arrow
Move focus and multiselect next data item	Shift+Down arrow
Move focus and multiselect previous data item (on left)	Shift+Left arrow
Move focus and multiselect next data item (on right)	Shift+Right arrow
Move focus to previous data item, without changing the current selection	Ctrl+Up arrow
Move focus to next data item, without changing the current selection	Ctrl+Down arrow
Move focus to previous data item (on left), without changing the current selection	Ctrl+Left arrow
Move focus to next data item (on right), without changing the current selection	Ctrl+Right arrow
Multiselect data item with focus	Ctrl+Space
Zoom in one level if zooming is enabled	= or +
Zoom out one level if zooming is enabled	- or _
Pan up if scrolling is enabled	Page Up
Pan down if scrolling is enabled	Page Down
Pan left in left-to-right locales/Pan right in right-to-left locales	Shift+Page Up
Pan right in left-to-right locales/Pan left in right-to-left locales	Shift+Page Down
Drill on data item, categorical axis label, or legend item when drilling is enabled	Enter

3.3.3.4 Map Region

Learn about keyboard shortcuts for the map region.

Action	Keys
Increase the zoom level by 1	= / +
Increase the zoom level by 2	Shift-= / Shift++
Decrease the zoom level by 1	-
Pan by 100 pixels	Arrow keys
Increase the rotation by 15 degrees	Shift+Right arrow
Decrease the rotation by 15 degrees	Shift+Left arrow
Increase the pitch by 10 degrees	Shift+Up arrow
Decrease the pitch by 10 degrees	Shift+Down arrow

3.3.3.5 Interactive Grid Region

Learn about keyboard shortcuts for the interactive grid region.

Interactive grid regions come with many keyboard shortcuts and keyboard-specific features to help the keyboard-only and power user. Specifically, they are designed to provide all functionality that is available with the mouse to keyboard users.

In addition, the grid containing the data provides two distinct modes of interaction:

Navigation and **Edit** mode. **Navigation** mode is designed for use when the grid is not currently editable and allows fast and easy navigation around the grid using keyboard. **Navigation** mode is the default and the only mode if the interactive grid is not editable. **Edit** mode is designed for use when the grid is currently editable, and keyboard behavior differs slightly.

The interactive grid region consists of multiple separate user interface elements, some of which handle tab stops differently depending on the element type and the expected keyboard semantics. For example, the toolbar and pagination controls are implemented as separate tab stops, while the grid view and icon view are implemented as a single tab stop where cursor key navigation is used to move around the view elements when the view has focus.

The following tables list available keyboard shortcuts in interactive grid regions. General shortcuts work in both navigation mode and in edit mode.

General Interactive Grid Keyboard Shortcuts

Action	Keys
Move to the next tab stop	Tab
Move to the previous tab stop	Shift+Tab
Toggle between row selection and cell selection	F8
Open the column header menu, when focus is on the column header	Enter or Space
Increase column width, when focus is on the column header	Windows: Ctrl+Right arrow Mac: Ctrl+Cmd+Right arrow
Decrease column width, when focus is on the column header	Windows: Ctrl+Left arrow Mac: Ctrl+Cmd+Left arrow

Action	Keys
Move column to next column position, when focus is on the column header	Shift+Right arrow
Move column to the previous column position, when focus is on the column header	Shift+Left arrow
Sort ascending on the current column, when focus is on the column header	Windows: Alt+Up arrow Mac: Option+ Up arrow
Sort ascending on the current column in addition to the existing search columns, when focus is on the column header	Windows: Shift+Alt+Up arrow Mac: Shift+Option+ Up arrow
Sort descending on the current column, when focus is on the column header	Windows: Alt+Down arrow Mac: Option+ Down arrow
Sort descending on the current column in addition to the existing search columns, when focus is on the column header	Windows: Shift+Alt+Down arrow Mac: Shift+Option+Down arrow
Move to next tab stop in column header menu, when column header menu is open	Tab
Move to previous tab stop in the column header menu, when column header menu is open	Shift+Tab
Close the column header menu, when the column header menu is open	Escape
Show help (if defined) for an item, when a single row is viewed and focus is on the column	Windows: Alt+F1 Mac: Option+F1

Navigation Mode Interactive Grid Keyboard Shortcuts

Action	Keys
Move around the grid, including column headers	Arrow keys
Move without changing the selection	Windows: Ctrl+Up arrow or Ctrl+Down arrow Mac: Shift+Option+Up arrow or Shift+Option+Down arrow
Select	Space
Toggle selection	Windows: Ctrl+Space Mac: Ctrl+Alt+Space
Range select and deselect multiple rows	Shift+Up arrow or Shift+Down arrow
Move one page up in the current column	Page Up
Move one page down in the current column	Page Down
Move to first cell in current row	Home
Move to last cell in current row	End
Move to the first cell in the grid	Windows: Ctrl+Home Mac: Option+Up arrow
Move to the last cell in the grid	Windows: Ctrl+End Mac: Option+Down arrow
Add a row after the last selected row, if grid is editable	Windows: Insert Mac: No equivalent
Delete the selected rows, if grid is editable	Delete
Move around the icons, while in icon view	Arrow keys

Action	Keys
Move to the first icon, while in icon view	Home
Move to the last icon, while in icon view	End
Switch from navigation mode to edit mode, if the interactive grid is editable	Enter or F2

Edit Mode Interactive Grid Keyboard Shortcuts

Action	Keys
Move to the next cell in the same row	Tab
Move to the previous cell in the same row	Shift+Tab
Move to the next cell in the same column	Enter
Move to the previous cell in the same column	Shift+Enter
Exit edit mode and return to navigation mode	Escape

3.3.3.6 Markdown Editor

Learn about keyboard shortcuts for the Markdown editor.

Action	Keys
Leave Markdown Editor and go to next tab stop	Ctrl+Shift+.
Leave Markdown Editor and go to previous tab stop	Ctrl+Shift+,

3.3.3.7 Rich Text Editor

Learn about keyboard shortcuts for the rich text editor.

Content Editing Shortcuts for the Rich Text Editor (TinyMCE)

Action	Keys
Bold	Windows: Ctrl+B Mac: Cmd+B
Italic	Windows: Ctrl+I Mac: Cmd+I
Underline	Windows: Ctrl+U Mac: Cmd+U
Select All	Windows: Ctrl+A Mac: Cmd+A
Redo	Windows: Ctrl+Y / Ctrl+Shift+Z Mac: Cmd+Y / Cmd+Shift+Z
Undo	Windows: Ctrl+Z Mac: Cmd+Z
Header 1	Windows: Alt+Shift+1 Mac: Ctrl+Option+1
Header 2	Windows: Alt+Shift+2 Mac: Ctrl+Option+2

Action	Keys
Header 3	Windows: Alt+Shift+3 Mac: Ctrl+Option+3
Header 4	Windows: Alt+Shift+4 Mac: Ctrl+Option+4
Header 5	Windows: Alt+Shift+5 Mac: Ctrl+Option+5
Header 6	Windows: Alt+Shift+6 Mac: Ctrl+Option+6
Paragraph	Windows: Alt+Shift+7 Mac: Ctrl+Option+7
Div	Windows: Alt+Shift+8 Mac: Ctrl+Option+8
Address	Windows: Alt+Shift+9 Mac: Ctrl+Option+9
Focus to toolbar	Windows: Alt+F10 Mac: Option+F10
Focus to element path	Windows: Alt+F11 Mac: Option+F11
Focus to contextual toolbar	Windows: Ctrl+F9 Mac: Ctrl+F9
Insert link	Windows: Ctrl+K Mac: Cmd+K

User Interface and Navigation Keyboard Shortcuts for the Rich Text Editor (TinyMCE)

Action	Keys
Execute command	Enter / Space
Focus on next UI element	Tab
Focus on previous UI element	Shift+Tab
Focus next control	Right arrow / Down arrow
Focus previous control	Left arrow / Up arrow
Open menu or toolbar menu button	Down arrow / Space
Open group toolbar button	Space
Open split toolbar button	Down arrow
Open the popup menu on split toolbar buttons	Shift+Enter
Open submenu	Right Arrow
Close submenu	Left Arrow / Esc
Close dialog	Esc
Close menu	Esc
Move focus back to editor body	Esc

4

Developing Accessible Apps

This section provides guidelines for creating an accessible app with Oracle APEX.

- [Themes](#)
Use Themes, including Universal Theme, to define the layout and style of applications.
- [Pages](#)
Enable easy navigation on Pages by configuring common attributes.
- [Regions](#)
Configure page regions to provide navigational information for users.
- [Form Pages and Items](#)
Configure form pages and items for users of assistive technology.
- [Buttons](#)
Use button labels and templates to support accessibility.
- [Testing Apps for Accessibility](#)
Learn about testing applications for accessibility.

4.1 Themes

Use Themes, including Universal Theme, to define the layout and style of applications.

Important:

Oracle recommends using Universal Theme for creating accessible apps.

This guide is written for use with the Universal Theme. This guide does not support customized themes directly. While some discussion applies generally to all apps, many discussions are specific to Universal Theme.

- [About Accessibility Features in Universal Theme](#)
Universal Theme is an accessibility-optimized theme for Oracle APEX.
- [About Accessible Theme Styles](#)
Mark a theme style as **accessibility tested** if it meets appropriate accessibility standards.
- [About Theme Roller](#)
Use Theme Roller to customize Universal Theme.

4.1.1 About Accessibility Features in Universal Theme

Universal Theme is an accessibility-optimized theme for Oracle APEX.

Universal Theme (also called Theme 42, or abbreviated as UT) was first introduced in Oracle Application Express 5.0 to improve out of the box usability and accessibility for Oracle APEX apps.

While you can create and use your own fully accessible custom themes in APEX, this guide only provides instructions for apps using Universal Theme.

Oracle strongly recommends migrating from older themes to Universal Theme. See the [Universal Theme migration guide](#) for further assistance.

4.1.2 About Accessible Theme Styles

Mark a theme style as **accessibility tested** if it meets appropriate accessibility standards.

Theme styles enable you to change the look and feel of an app without switching to an entirely different theme.

Oracle recommends using the default Oracle APEX theme style Vita for best accessibility.

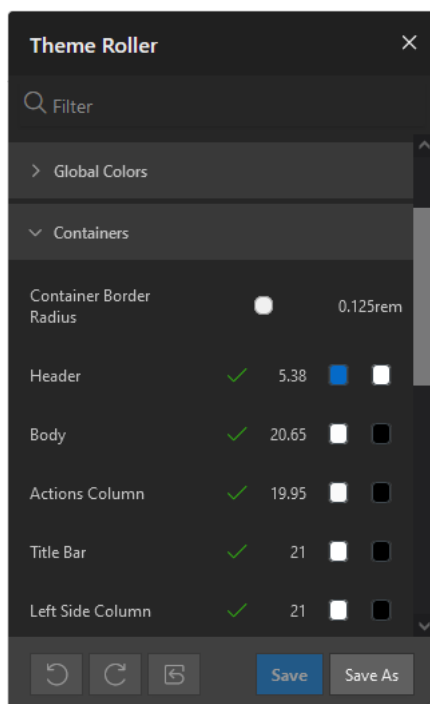
You can mark theme styles in APEX as **accessibility tested**. Because the underlying CSS of a theme style affects the overall accessibility of an app, you can mark a theme style as accessibility tested if you are confident that it meets the appropriate accessibility standards that you or your organization must adhere to. The Vita theme style is accessibility tested.

4.1.3 About Theme Roller

Use Theme Roller to customize Universal Theme.

Theme Roller is a developer tool in the application runtime for analyzing and declaratively adjusting the style settings of Universal Theme without needing to know the underlying CSS.

Some color combinations cause accessibility issues due to a lack of contrast. To help prevent this, Theme Roller has automatic color contrast checking that analyzes your app's compliance with WCAG 2.0 contrast standards. Theme Roller notifies you if a setting passes or fails this contrast check.



A check mark appears next to color combinations for settings to indicate that the color contrast defined meets the minimum required ratio. The ratio is also included next to the check mark.

To learn more about minimum color contrast, see [WCAG 2.1 Understanding Success Criterion 1.4.3: Contrast \(Minimum\)](#).

4.2 Pages

Enable easy navigation on Pages by configuring common attributes.

- [Page Titles](#)
Page titles help users understand the purpose of a page.
- [Page Modes](#)
Oracle APEX supports multiple page mode types.
- [Cursor Focus](#)
Cursor focus specifies where the cursor appears for the user on page load.
- [Reload on Submit](#)
Reload on Submit controls how attributes display on the page.

See Also:

- Right Pane of Page Designer (Property Editor) in *Oracle APEX App Builder User's Guide*
- Accessing Property Editor in *Oracle APEX App Builder User's Guide*

4.2.1 Page Titles

Page titles help users understand the purpose of a page.

A good page title helps users of assistive technology understand the purpose of the page. Since the page title is usually the first thing a user of a screen reader hears when loading the page, every page title should be meaningful and unique.

Consider the following example. The title of a Form page should include some information about the record the user is currently editing, such as "Create / Edit Customer: [customer name]." In Page Designer, select the page in the Rendering tree and do the following in the Property Editor:

- To replace P1_CUSTOMER_NAME with the page item containing the item that best identifies the current record, in the **Identification – Title** field enter `Create / Edit Customer: &P1_CUSTOMER_NAME.`, with the period included.
- Form pages have a Fetch Row process that fetches the row in the Execution Options. By default, this is set to fire after the page header (**After Header**). Change this to fire **Before Header**.



Tip:

You can review all page titles in an app with the **Page Attributes** tool. You should review all titles for accuracy and meaning, especially different pages with the same title and null page titles.



See Also:

- Controlling When a Page Process Executes in *Oracle APEX App Builder User's Guide*
- Accessing Cross Page Utilities in *Oracle APEX App Builder User's Guide*

4.2.2 Page Modes

Oracle APEX supports multiple page mode types.

You can select from the following page mode types:

- **Normal**
- **Modal Dialog**
- **Non-Modal Dialog**

All of these page types can be used and are coded to the accessibility standards.

For example, modal dialog pages follow guidelines for pages including constraining keyboard focus inside of the dialog and ensuring screen reader users only hear

information about the current modal dialog page (and not the base page where the modal dialog page was launched).

4.2.3 Cursor Focus

Cursor focus specifies where the cursor appears for the user on page load.

The **Navigation – Cursor Focus** attribute specifies where the cursor appears for the user on page load. You can select from the following options:

- **Do not focus cursor**, which focuses on nothing on page load
- **First item on page**, which focuses the cursor in the first field on the page on page load.

You should use **First item on page** with careful consideration, as setting focus unexpectedly can easily disrupt users. Only set focus to the first item if you are certain users need to interact with that first item—for example, if the first item is a required field on a wizard page. Otherwise, you should use **Do not focus cursor**.

4.2.4 Reload on Submit

Reload on Submit controls how attributes display on the page.

Reload on Submit controls how errors display on the page. You can select from the following options:

- Always
- Only for Success

Oracle recommends using **Only for Success**. When **Only for Success** is selected, Oracle APEX renders errors on the client-side without reloading the entire page. This option is faster for all users, and if there are errors, screen reader users are immediately notified and read the error information because the error notification is using the ARIA role of *alert*.



Note:

Tabular form regions can only be set to Always. You cannot use this client-side error display on tabular form pages.



See Also:

[Dealing with Validation Errors](#)

4.3 Regions

Configure page regions to provide navigational information for users.

- [Page Structure](#)
Page structure helps users navigate to the content they're interested in.

- [Report Regions](#)
Report regions are more accessible if you define the report row header columns .
- [Interactive Grid Regions](#)
Interactive grids are fully accessible with a keyboard.
- [Interactive Report Regions](#)
Interactive reports benefit from data table labels and row-specific text for the edit icons.
- [Classic Report Regions](#)
Classic reports benefit from data table labels and row-specific text for the edit icons.
- [JET Chart Regions](#)
JET Chart Regions have multiple settings to improve accessibility.

4.3.1 Page Structure

Page structure helps users navigate to the content they're interested in.

Users of assistive technology benefit greatly if your APEX page has a well-defined page structure. If a page is well-defined and includes excellent **Heading** and **Landmark** structures, users of assistive technology can easily scan the page to orient themselves, and then immediately navigate to the content of interest.

This section describes how to create a well-defined page structure in APEX.

- [Landmarks](#)
Learn about landmark roles and labels.
- [Headings](#)
Learn about headings and heading levels.

4.3.1.1 Landmarks

Learn about landmark roles and labels.

Landmarks provide a powerful way to identify the structure of an APEX application. They should be used to complement, not replace, Heading structure. Defining a landmark involves defining a landmark role and a unique landmark label:

- Landmark Roles:
 - allow developers to define the purpose of specific sections on a page
 - include **Navigation**, **Search**, **Main**, and similar
- Landmark Labels:
 - are announced, along with the role, to screen reader users
 - are not always required depending on the circumstances

For example, a `search` landmark with an `application` label is announced as "Application search".

With landmarks, you do not have to define the hierarchy level like you do with headings because the level is automatically derived from the page structure. Correctly-implemented landmarks can be much more powerful than headings.

Example 4-1 Simple Landmarks

This example demonstrates simple landmarks, showing only the landmark role:

- Banner
- Navigation
- Main
- Complementary
- Content Information

Example 4-2 Nested Landmarks

This example demonstrates nested landmarks. It shows the landmark role, with the landmark label in brackets:

- Banner
- Navigation (Breadcrumb)
 - Search
 - Region (Employees Report)
 - Region (KPIs for Quarter)
 - Region (Overdue Tasks)
 - Region (Employee Performance Chart)
- Content Information
- [Landmark Roles](#)
Different landmark roles have different purposes.
- [Landmark Support in APEX](#)
Learn about landmark support in APEX.

4.3.1.1.1 Landmark Roles

Different landmark roles have different purposes.

Banner

A banner landmark identifies application-oriented content at the beginning of each page within your APEX application, typically as the **header** of the page. Within APEX, this is usually everything contained within the top bar, but this can vary depending on your application, application theme, or template. For example, the banner could include the application logo, name, and navigation bar controls.

Pages usually only have one banner landmark. The most common pattern for APEX pages is for this banner landmark to come from the page template. Manually defining page regions as banner regions in APEX is rare.

Complementary

A complementary landmark is a supporting section of a page, at a similar level in the DOM hierarchy as the main content, but still meaningful when separated from the main content.

Content Info

A content info landmark identifies common information at the bottom (or **footer**) of each application page.

Pages usually only have one content info landmark. The most common pattern for APEX pages is for this content info landmark to come from the page template.

Manually defining page regions as content info regions in APEX is rare.

Form

A form landmark identifies a region that contains a collection of elements, such as page items and buttons.

Main

A main landmark identifies the primary content of the page.

Each page should have only one main landmark. The most common pattern for APEX pages is for the main landmark to come from the page template. There is usually no need to manually define individual regions as main landmarks.

Navigation

Navigation landmarks provide a way to identify groups (for example, lists) of links that are intended to be used for application or page content navigation.

Region

A region landmark is a perceivable section containing content relevant to a specific purpose and sufficiently important that users likely want to navigate to the section easily, and have the section listed in a summary of the page. This is the most common landmark type for APEX regions.

You can use a region landmark to identify content that other named landmarks do not appropriately describe.

Search

A search landmark contains a collection of elements (usually page items and buttons) that create search functionality for content in the application.

4.3.1.1.2 Landmark Support in APEX

Learn about landmark support in APEX.

In APEX, landmarks are defined at different places in the framework, most commonly in the page template, within specific components, and at the region level. As an APEX developer, defining your regions with appropriate landmarks is essential to achieve good page landmark structure.

When working with regions, you have great flexibility in how these regions are (or aren't) included as landmarks on the page. This includes the following options in Page Designer:

- **Use Landmark:** Toggles whether to expose the region as a landmark. You may not want to expose the region as a landmark if, for example, it is just used for layout purposes in Page Designer, and there would be no value in exposing it to a screen reader user.
- **Landmark Type:** Define the type of landmark. This can either be the template default or custom for the current region.
- **Landmark Label:** When labeling landmarks, you can use the region title (default), or define a custom Landmark Label for the region.

 Note:

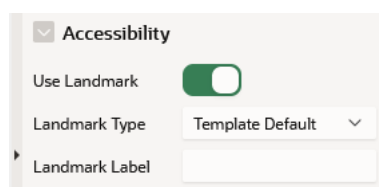
If no label is defined, the region title is used as the landmark label.

It is important to provide good labels that truly improve user experience. A good label is:

- Unique on a page
- Not too long (preferably 3 words or less)
- An accurate description of the purpose of the region contents. For example, `Edit Employee Smith`, or `Useful Links`.

 Tip:

Do not use the landmark role in the label. For example, the label for a navigation landmark should be `Application`, rather than `Application Navigation`, to avoid screen readers announcing navigation twice.

 **Note:**

If you do not see these options for the regions on a page, it is because the current theme's region templates do not define the `#REGION_LANDMARK_ATTRIBUTES#` substitution string under Template, Definition.

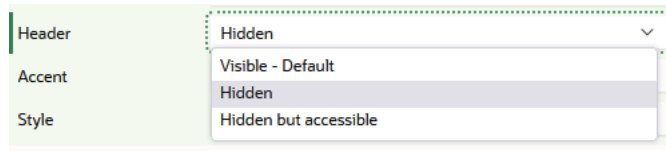
Theme developers should define the **Default Landmark Type** and **Region Title HTML ID** properties in the template in order to fully support landmarks. If you use Universal Theme, ensure you refresh the theme to the latest version. The latest version of Universal Theme defines these attributes for all region templates.

 See Also:

Runtime Developer Toolbar** in *Oracle APEX App Builder User's Guide*.

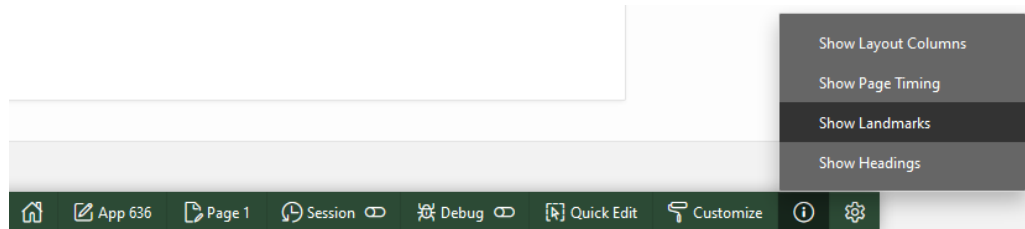
Interaction with Template Options

The region template option **Header** influences a region's landmark definition. If this template option is set to `Hidden`, APEX assumes you do not want to expose this region title to assistive technology and hides the landmark.



Visualizing Page Structure with the Developer Toolbar

The Developer Toolbar includes an option to visualize the landmarks included on a page, to help in testing whether good landmarks have been defined.



Selecting **Show Landmarks** creates an overlay in the runtime of the page that shows the landmarks defined on the page, including their type and label. Here are some things to look out for when reviewing the page landmarks:

- Define **Banner**, **Content Info**, and **Main** landmarks once per page. If there is more than one of these on a page, change the landmark type.
- Check that labels meet guidelines for good labels: unique on a page, not too long, and not including the landmark role in the label.
- Define **Banner**, **Main**, **Complementary**, and **Content Info** at the top level. These landmarks should not be contained inside any other landmarks on the page. If any of these are contained inside other landmarks, you should move them or change the landmark type.
- Ensure each landmark type is appropriate for the content. For example, form items are in a **Form** landmark and search functionality is in a **Search** landmark.
- All perceivable content, including items, buttons, regions, and menus is contained in a landmark to ensure users cannot accidentally miss content. Content outside of a landmark region is an accessibility issue. You can fix the issue by re-positioning the content inside an appropriate landmark.

4.3.1.2 Headings

Learn about headings and heading levels.

Screen reader users may review all the heading elements on a page to gain an understanding of page structure. Defining a heading involves defining a heading level and heading text. A heading level is indicated by **H**, followed by the level number. For example, a third-level heading is **H3**.

Heading levels should be hierarchically nested, with one **H1** on a page to uniquely identify the page contents, followed by any **H2** (and subsequent) elements.

H1 Headings

Include a breadcrumb region on your pages to create a unique H1. You can use the Create Application Wizard to generate breadcrumbs automatically based on the hierarchy of pages defined in the wizard, or you can configure the breadcrumbs in your app manually. By default, the last breadcrumb entry on a page is the H1.

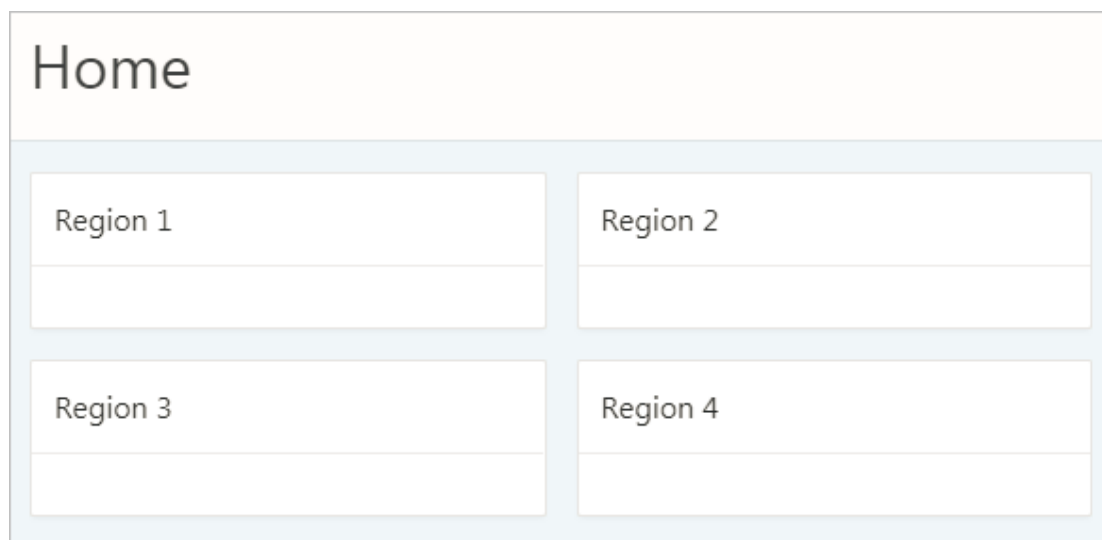
Alternatively, use the Hero region template in Universal Theme. This region template renders the region title as the H1 text on the page by default, which is useful if your current app layout and design does not warrant breadcrumbs.

**Tip:**

Only use one Hero region per page to avoid creating more than one H1 element.

H2 Headings

After H1, most region templates in Universal Theme render other content on the page with H2 around the region title



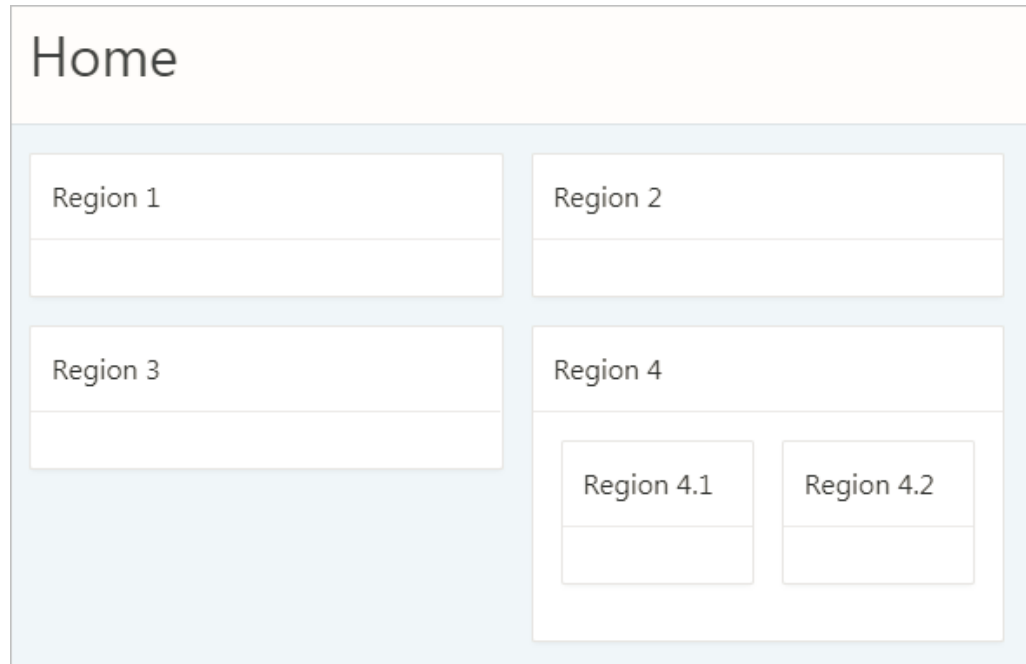
H2s are appropriate for non-nested regions on a page, such as the following example heading structure:

```
H1: Charts
  H2: Region 1
  H2: Region 2
  H2: Region 3
  H2: Region 4
```

This semantic structure helps all users scan the current page, both visually, using the eyes, and audibly, using a screen reader.

H3 and Other Sub-Region Headings

Universal Theme does not automatically support headings beyond H2, such as when you have sub-regions nested within an H2 region.



By default, APEX applies H2s to nested sub-regions (regions 4.1 and 4.2 in the above example) in the Standard region template. This creates an accessibility problem because screen reader users may incorrectly believe that these sub-regions are of similar importance to the parent region or that the regions are not ancestral to the parent region.

```
H1: Charts
  H2: Region 1
  H2: Region 2
  H2: Region 3
  H2: Region 4
    H2: Region 4.1
    H2: Region 4.2
```

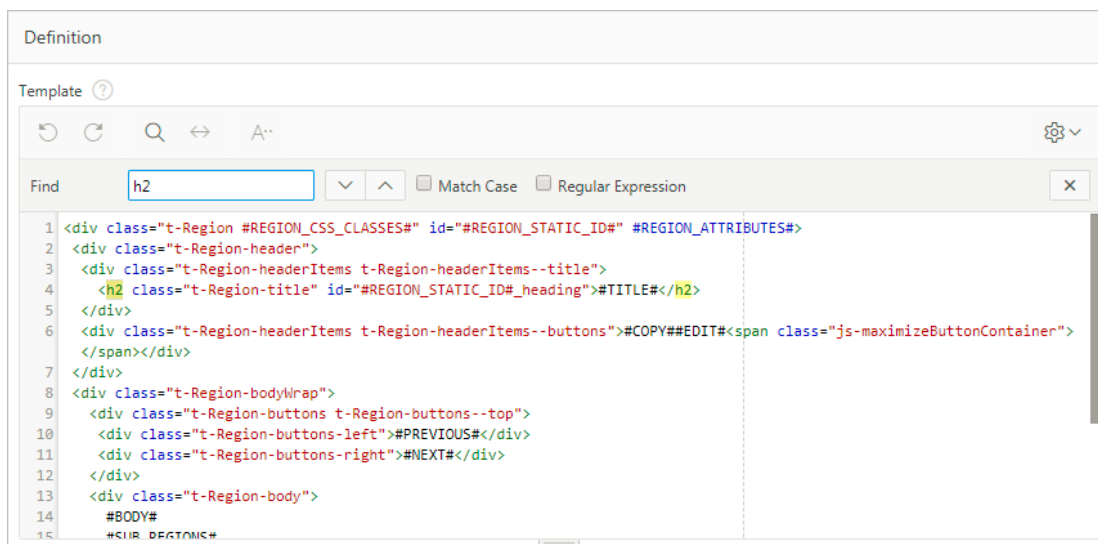
To properly structure these sub-regions, you must duplicate the Standard region template, then use the Code Editor to manually replace H2 tags with H3s in the template definition.



Tip:

To copy a region, see *Creating a New Template in Oracle APEX App Builder User's Guide* and select the **As a Copy of an Existing Template** option.

Changing the sub-regions to H3s only affects the structure of the element, not the style, which is what we want for accessibility.



This results in the following heading structure:

```

H1: Charts
  H2: Region 1
  H2: Region 2
  H2: Region 3
  H2: Region 4
    H3: Region 4.1
    H3: Region 4.2

```

! Important:

Be aware of the following issues when copying the Standard region template:

- While a copied template keeps the theme subscription, allowing you to refresh the theme after you upgrade APEX, the copied template does not upgrade automatically. You may need to manually change or re-copy the template after the refresh if you have made markup changes in that template.
- You must manually add an additional template for every deeper heading level in your app (H3, H4, H5...). We plan to make it easier to control the heading level without requiring an additional template in a future APEX release.

Region Titles

You should define meaningful titles for most region types because Universal Theme uses the region title as the heading text for the region.

Descriptive titles are essential for accessibility. By default, titles are visible on screen. However, sometimes you may not want a visible title for a region, such as when sighted users can quickly discern the region's content from the surrounding page content. In this cases, you must still provide heading text for screen reader users. To do this, select the region. In Property Editor, open the Template Options, and set the Header attribute to **Hidden but accessible**.

**Tip:**

You can review and update all region titles in an app with the **Edit Region Titles** page. You should review all titles for accuracy and meaning, especially regions with the same title on the same page, which may be an issue depending on the context.

**See Also:**

Accessing Page Specific Utilities in *Oracle APEX App Builder User's Guide*

4.3.2 Report Regions

Report regions are more accessible if you define the report row header columns .

Classic reports, Interactive reports, and Interactive Grids all provide a way of defining the report row header column(s), which greatly improves screen reader experience. Row header columns are defined using the **Value Identifies Row** column attribute.

Define the column value, or combination of column values, that help identify the row of data. For person data, this could be the first and last name; for customer data, this could be the customer name, and so on.

When a screen reader user navigates to different rows of the report, the column, or columns, defined as identifying the row are announced as the row header as soon as the users moves to the new row. If **Value Identifies Row** is not defined, it is much harder for these users to orient themselves when moving around the report.

4.3.3 Interactive Grid Regions

Interactive grids are fully accessible with a keyboard.

Introduced in Oracle Application Express 5.1, interactive grids combine powerful reporting with grid-based editing in a single component. Users can access all interactive grid functionality with just a keyboard, enabling all users to navigate around the grid and edit their data using the keyboard. For a complete list of keyboard support in interactive grids, see [Interactive Grid Region](#).

4.3.4 Interactive Report Regions

Interactive reports benefit from data table labels and row-specific text for the edit icons.

Data Table Accessible Label

Oracle APEX generates the following SUMMARY text for the rendered data table:

```
Region = [region title], Report = [current saved report name], View =  
[current View], Displayed Rows Start = [current start of rows displayed,
```

Displayed Rows End = [current end of rows displayed], Total Rows = [total number of rows]

This text identifies the data table, and therefore provides a great deal of contextual information to users of assistive technology.

Edit Icon Link Text

When creating a Report and Form, APEX generates the links between the two pages to allow you to edit specific records. The link text generated for all rows is `Edit` by default. You should make this text a meaningful label so that screen reader users (who listen to all the links on the current page) can understand the purpose of the link and where it takes them. The link text should be specific to the current row in the report. For example, for a report of customers, the link text could be `Edit Customer: SMITH`.

You can define the link text of an interactive report region by selecting the **Attributes** node, then in the Property Editor expand the **Link** column, set **Link Column** to **Link to Single Row View** or **Link to Custom Target**, and update the alt text in the **Link Icon** field.

Because there is no visible link text used, the link uses the inline image's alt text as the text for the link. Note that you can use a substitution string such as `#CUSTOMER_NAME#` to target the column that best identifies the current row.



See Also:

Editing Interactive Report Attributes in *Oracle APEX App Builder User's Guide*

4.3.5 Classic Report Regions

Classic reports benefit from data table labels and row-specific text for the edit icons.

Data Table

By default, Oracle APEX generates **SUMMARY** text equal to the region title. This identifies the data table to users of assistive technology.

Edit Link Text

When creating a Report and Form, APEX generates the links between the two pages to allow you to edit specific records. The link text generated for all rows is `Edit` by default. You should make this text a meaningful label so that screen reader users (who listen to all the links on the current page) can understand the purpose of the link and where it takes them. The link text should be specific to the current row in the report. For example, for a report of customers, the link text could be `Edit Customer: SMITH`.

You can define the link text of an interactive report region by selecting the **Attributes** node, then in the Property Editor expand the **Link** column, set **Link Column** to **Link to Single Row View** or **Link to Custom Target**, and update the alt text in the **Link Icon** field.

Because there is no visible link text used, the link uses the inline image's alt text as the text for the link. Note that you can use a substitution string such as `#ENAME#` to target the column that best identifies the current row.

4.3.6 JET Chart Regions

JET Chart Regions have multiple settings to improve accessibility.

Oracle Application Express 5.1 introduced charts based on the Oracle JET Data Visualization components (JET charts). Releases prior to 5.1 used charts based on AnyChart, where there were many accessibility issues.

For more information about accessibility of Oracle JET Data Visualization components including support for keyboard and touch, see the [Oracle JET documentation](#).

The following sections detail some chart settings that are particularly useful for ensuring chart accessibility.

Chart Title

The chart title displays in the chart region next to the chart. It is also the accessible label for the chart page section. In Page Designer, the chart title is defined in the chart Attributes node.

If there is only one chart displayed within a region, consider hiding the region title to avoid visual duplication. You can hide the region title by changing the Header region appearance setting in the Template Options dialog to **Hidden but accessible**. This removes the region title from the page but preserves the heading for assistive technologies (if users scan the current headings on the page to orientate themselves, they can perceive this title).

Tooltip

The tooltip displays when the mouse hovers over a specific chart section or the keyboard has focus on this section. In addition, the tooltip is the accessible label for that section of the chart. For example, when screen reader users navigate around the chart sections, this tooltip is announced.

To display the tooltip, select the chart Attributes node in the Rendering tree; in the Property Editor, set the Show field to **Yes**; and finally, select what information to include in the tooltip. You can define this in two ways:

- Adjust the other attributes displayed in the Tooltip group in Page Designer. This allows you to control display of Series Name, Group Name, Value, and Label information where applicable for your chart. For example, if the chart only has one series, then you may want to exclude the series name in the tooltip, so you would set Show Series Name to **No**.
- For a highly customized tooltip, set a column in your database table to be used as the source for the tooltip. You can configure this on the individual Series node nested under the chart in the Rendering tree.

Automatic Refresh Functionality

Developers can define an **Automatic Refresh, Interval** in the chart attributes which causes the chart region to automatically refresh according to the interval defined. Oracle recommends avoiding this functionality for accessibility purposes since this type of automatic refresh is very disruptive for users of assistive technology.

4.4 Form Pages and Items

Configure form pages and items for users of assistive technology.



Note:

The following overview is not a comprehensive list of all item types in Oracle APEX . This section only cites items that have specific settings or issues related to accessibility.

- [Form Page Attributes](#)
Define meaningful labels for all items.
- [Select Lists and Radio Groups](#)
Use dynamic actions and cascading LOVs to control page reloads and redirects.
- [Display Image Alternative Text](#)
Configure alternative text to describe images.

4.4.1 Form Page Attributes

Define meaningful labels for all items.

Item Labels

Define meaningful labels for all items. Users of assistive technology require meaningful labels to understand and interact with a field.



Tip:

You can review all items in an app with the **All Page Items** page. You should review all labels for accuracy and meaning, especially items with null labels and items with the same label on the same page.



See Also:

Viewing Item Utilities in *Oracle APEX App Builder User's Guide*

Value Placeholder

Oracle recommends against using just a value placeholder as the only label for an item because it is insufficient for accessibility.

To hide a visible label from display for design reasons, Oracle recommends using a Hidden label template in conjunction with a label.

Item Templates

Universal Theme includes the following item templates to control how the label renders for the item:

Hidden

Hides the label from display but preserves it in the markup for use by assistive technologies.

Use this template to preserve accessibility rather than not providing a label.

Optional / Optional - Above

Labels the field optional without any further decoration.

Required / Required - Above

Labels the field required and displays an asterisk adjacent to the label.

For accessibility, Oracle APEX ignores the asterisk and appends hidden text "Value Required" to the label. Assistive technology announces this label as "*Label* (Value Required)" so that the user can immediately tell if the field is required.

Additionally, APEX adds "Value Required" to the end of the label rather than the beginning, so that screen readers users can still search for a specific field by label name.

Value Required

Oracle does not recommend use of the Value Required attribute for accessibility purposes. If the Value Required attribute is set to **Yes** and the item's label template is **Required**, APEX duplicates reporting of the required state to Assistive Technologies (see the ACR to learn more).

If you have a simple required field, use a **Not Null** style validation on the page item.



See Also:

- [Dealing with Validation Errors](#)
- Validating User Input in Forms in *Oracle APEX App Builder User's Guide*

4.4.2 Select Lists and Radio Groups

Use dynamic actions and cascading LOVs to control page reloads and redirects.

With the Page Action on Selection attribute, you can define select lists and radio groups to redirect or submit the page when the value of the control changes. However, this behavior could be disruptive to users who may not expect a page reload or change of context. There are a few ways this can be handled in a more accessible way.

Instead of redirecting or submitting the page automatically, you could:

- Use dynamic actions or cascading LOVs to achieve similar functionality
- Add a button next to the page item that handles the redirect/submit explicitly

Consider the following example, where you have a select list on a page to filter a report region, and you submit the page with this attribute in order to update the report and show the filtered results. Instead, use a Dynamic Action on Change for the select list, and then define an action to refresh the report region. For further examples, see the **Sample Dynamic Action** app, available to install from the Gallery.

If redirecting or submitting the page on a change cannot be avoided, you must ensure the label for the item lets the user know this will happen so that the change of context is expected. For example, a label of `Select Employee (Reloads Page)` indicates that making a selection for this item will reload the content.



See Also:

Installing Apps from the Gallery in *Oracle APEX App Builder User's Guide*

4.4.3 Display Image Alternative Text

Configure alternative text to describe images.

The display image item type enables you to display alternative text with an image on a form page, which is essential to users of assistive technology.

Alternative text settings for the display image item type relevant to accessibility include:

Alternative Text Column

Displayed when Based On is set to **BLOB Column Specified in Item Source**

Defines the column that stores the alternative text for the image.

Alternative Text

Displayed when Based On is set to **Image URL stored in Page Item Value**

Defines alternative text to describe the image. You can make the text dynamic by using substitution syntax, such as `&P1_ALT_TEXT..`. Make sure to include the period in the substitution syntax.

4.5 Buttons

Use button labels and templates to support accessibility.

Label

The label is crucial for button accessibility. Assistive technologies such as screen readers announce the button label, and users with speech recognition software can speak the name of the label to activate the button. Ensure your button labels are clear, unique, and meaningful.

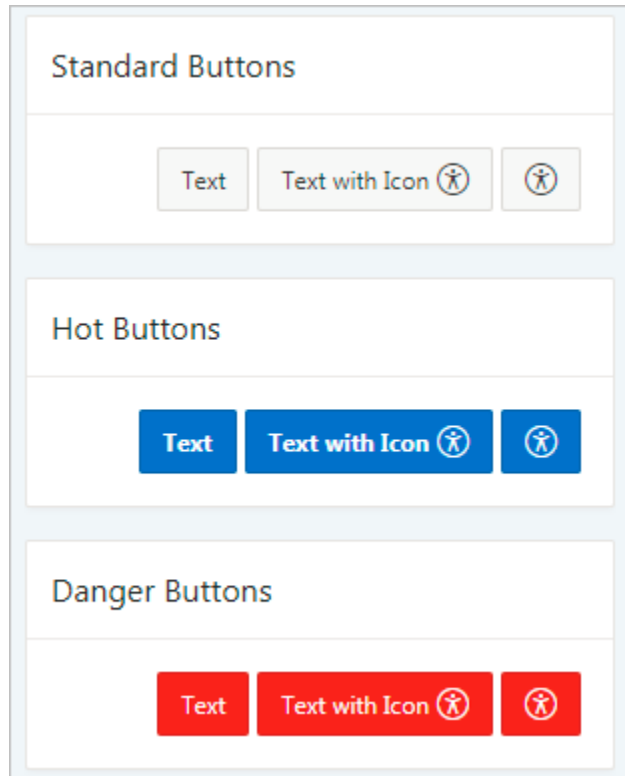
Button Template

Buttons can be based on the following templates:

- Icon
- Text
- Text with Icon

All of these are accessible. Labels for icon-based buttons are hidden from display, but they still exist in the markup for screen readers.

Template Options



Be aware that in the Template Options dialog, the **Type** option changes the button design in ways that can affect accessibility. For example, the Danger type colors the button background red and the foreground text white. Conveying information by color alone is an accessibility issue, so only use the Type template options with this potential problem in mind. For example, the Danger type can still be accessible if the button text clearly indicates the purpose of the button (as opposed to only an exclamation point).

Tip:

You can review button labels in an app with the **Page Buttons** page. You should review all labels for accuracy and meaning, especially buttons with the same label on the same page.

See Also:

Accessing Page Specific Utilities in *Oracle APEX App Builder User's Guide*

4.6 Testing Apps for Accessibility

Learn about testing applications for accessibility.

APEX is designed, built, and tested for accessibility. Oracle publishes an Accessibility Conformance Report to document known issues and gaps in the framework. When building applications with APEX, it is your responsibility to ensure the accessibility of your application by:

- taking time to understand the guidelines you are required to follow
- understanding the known defects with APEX
- conducting your own testing
- [About Accessibility Testing](#)
Develop a full testing strategy for your applications.
- [About Accessibility Checks in Advisor](#)
Use the Advisor to perform accessibility checks on a single page or on an entire application.
- [Running Advisor on a Single Page](#)
Perform accessibility checks (and others) on a page in an app.
- [Running Advisor on an Entire Application](#)
Perform accessibility checks (and others) on an entire app.

4.6.1 About Accessibility Testing

Develop a full testing strategy for your applications.

Testing for accessibility is wide and varied, and devising a full testing strategy is beyond the scope of this document. However, in essence, a full testing strategy should include:

- testing throughout the development cycle, rather than just at the end
- making testing the responsibility of all developers on your team and an integral part of your design and development process
- testing with people with different abilities and assistive technologies
- paying particular attention to any non-standard content included in your app, like plug-ins or 3rd party libraries

APEX includes a built-in tool called the Advisor, which developers can run as part of quality assurance of an application. The Advisor includes a set of accessibility checks to highlight possible common accessibility problems, such as page items with no labels or page regions with no row headers defined. Developers should review and fix these problems as part of their ongoing development process. We recommend fixing issues identified by the Advisor before performing other accessibility testing, such as testing with assistive technologies.

4.6.2 About Accessibility Checks in Advisor

Use the Advisor to perform accessibility checks on a single page or on an entire application.

The Advisor performs the following accessibility checks when you run it. These checks identify some common configuration errors that have a negative impact on accessibility.

Theme Style tested for accessibility

Theme Styles that have not been tested for accessibility may contain more issues, such as insufficient color contrast.

Page has page title

Meaningful page titles help users understand the content and purpose of the current page. (Note: Global pages, and pages with no regions are excluded from this check.)

Region has Row Header

Regions that support row headers should have a column with the **Value Identifies Row** attribute set to Yes.

Page item has label

The item should have a defined label. For example, only defining the Value Placeholder text is not sufficient in labelling an item for accessibility.

Page item does not cause an unexpected context change

Some page item settings can cause an unexpected change of context for the user, such as select lists that submit the page after a value is selected.

Consider the following ways to retain the context of the page:

- Remove page submit or redirect behavior from page items, and replace the functionality with Dynamic Actions or Cascading LOVs (if you need to stay on the same page).
- Notify users what happens when they click an item. For example, adding "Launches new page" in the label.

Display Image item has image ALT text defined

Display Image page items must provide text or a column (depending on the Based On setting) to serve as the image's alternative text. This is important for accessibility, as it is the only way for some users to perceive the content of the image.

4.6.3 Running Advisor on a Single Page

Perform accessibility checks (and others) on a page in an app.

To run Advisor on a single page:

1. Navigate to the appropriate page.
2. Click the **Utilities** menu and select **Advisor**.
Advisor appears.
3. Under Checks to Perform, review the selected options. Select and deselect options as appropriate.
4. Click **Perform Check** at the top of the page.
A Results page appears.
5. To alter the existing preferences:
 - a. Under Filter Result, deselect the appropriate options.
 - b. Click **Apply Filter**
6. To re-run the Advisor and see if reported problems are fixed, click **Perform Check**.

4.6.4 Running Advisor on an Entire Application

Perform accessibility checks (and others) on an entire app.

To run Advisor on an entire application:

1. Navigate to the appropriate application:
 - a. On the Workspace home page, click the **App Builder** icon.
The App Builder home page appears.
 - b. Select the application.
The Application home page appears.
2. Click **Utilities**.
3. Click **Advisor**.
The Advisor page appears.
4. Under Checks to Perform:
 - a. Expand Checks to Perform.
 - b. Review the selected options. Select and deselect options as appropriate.
5. In Pages(s), enter a comma separated list of pages in the field provided. To check all pages, leave this option blank.
6. Click **Perform Check** at the top of the page.
A Results page appears.



Tip:

Current Advisor settings are used next time the check is performed.

7. To alter the existing preferences:
 - a. Under Filter Result, deselect the appropriate options.
 - b. Click **Apply Filter**.
8. To run Advisor again, click **Perform Check**.

5

Open Bugs and Known Issues

Review known issues in Oracle APEX in the Accessibility Conformance Report (ACR).

The ACR (formerly known as the VPAT) includes details of all known issues with APEX. Please go to [ACRs for Oracle Database related products](#), then select **Database Products** or **Open All**, then find the relevant version of APEX.

The ACR lists exceptions in the product, along with the corresponding bug number. Bugs are categorized as one of the following:

- Affecting the Development Environment
- Affecting Applications
- Affecting Documentation

6

Further Reading

Additional resources for the assessment and enhancement of accessibility in Oracle APEX apps include the following documents.

- [Oracle's Accessibility Program](#) - General information about accessibility at Oracle.
- [Plug-ins](#) - Plug-ins enable developers to readily extended with custom item types, region types, processes, and dynamic actions. Once defined, plug-in based components are created and maintained very much like standard APEX components.
- [Universal Theme Sample App](#) - Sample app hosted on apex.oracle.com with information about Universal Theme, including a Migration Guide to move from older themes.

Index

A

accessibility
 at Oracle, [2-2](#)
 definition, [2-1](#)
 developing apps for, [4-1](#)
 on the web, [2-1](#)
 recommended system configuration, [3-1](#)
 resources, [6-1](#)
 testing apps for, [4-21](#)
Accessibility Conformance Report, [2-2](#), [4-21](#), [5-1](#)
 See also ACR
Accessible Rich Internet Applications
 See WAI-ARIA support
ACR, [2-2](#), [4-17](#), [5-1](#)
Advisor
 about, [4-21](#)
 running on entire app, [4-23](#)
 running on single page, [4-22](#)
alt text
 See images
attributes
 Link to Custom Target, [4-14](#), [4-15](#)
 Link to Single Row View, [4-14](#), [4-15](#)
 Page Action on Selection, [4-18](#)
 Reload on Submit, [4-5](#)
 Value Identifies Row, [4-14](#)

B

browser
 recommended, [3-1](#)
bugs
 See ACR
buttons
 labeling, [4-19](#)
 templates, [4-19](#)

C

Chart Region, [3-12](#)
charts, [4-16](#)
Code Editor, [3-7](#)
components
 accessing with keyboard

components (*continued*)

 See keyboard shortcuts component-specific
CSS Calendar, [3-9](#)
cursor focus, [4-5](#)

D

Date Picker, [3-8](#)
development environment
 accessing with keyboard, [3-5–3-7](#)

F

forms, [4-17](#)
 accessing using JAWS, [3-2](#)
 attributes in, [4-17](#)
 dynamic actions in, [4-18](#)
 regions in, [3-2](#)

G

general environment
 accessing with keyboard, [3-4](#)
 accessing with keyboards, [3-5](#)

H

headings, [3-2](#), [4-10](#)
help
 field-level, [3-3](#)

I

images, [4-19](#)
 alternative text for, [4-19](#)
items
 labeling, [4-17](#), [4-18](#)
 templates for, [4-17](#)
 value placeholders for, [4-17](#)

J

JAWS
 accessing forms, [3-2](#)

K

keyboard shortcuts,
 chart regions, [3-12](#)
 code editor, [3-7](#)
 component-specific, [3-8](#), [3-9](#), [3-12](#), [3-13](#),
 [3-15](#)
 CSS Calendar, [3-9](#)
 Date Picker, [3-8](#)
 development environment, [3-5–3-7](#)
 field-level help, [3-3](#)
 general, [3-4](#), [3-5](#)
 interactive grid, [3-13](#)
 map regions, [3-13](#)
 Markdown editor, [3-15](#)
 page designer, [3-6](#)
 rich text editor, [3-15](#)
 SQL Commands, [3-7](#)
 known issues
 See ACR

L

landmarks, [4-6](#)

M

Map Region, [3-13](#)
 markdown editor, [3-15](#)

P

Page Designer, [3-6](#)
 pages
 cursor focus, [4-5](#)
 forms
 See forms
 headings, [3-2](#)
 modes, [4-4](#)
 Reload on Submit, [4-5](#)
 running accessibility checks on, [4-22](#)
 structure of, [3-2](#), [4-6](#), [4-10](#)
 titles, [4-4](#)
 PDFs
 accessibility tags in, [3-2](#)
 recommended software, [3-1](#)

R

radio groups, [4-18](#)
 regions
 developing, [4-5](#)

regions (*continued*)

 hero, [4-10](#)

reports

 classic reports, [4-14](#), [4-15](#)
 interactive grids, [3-13](#), [4-14](#)
 interactive reports, [4-14](#)

rich text editor, [3-15](#)

S

screen readers

 edit icon link text for, [4-14](#), [4-15](#)
 fixing validation errors, [3-3](#)
 item errors, [3-4](#)
 recommended, [3-1](#)
 report row headers for, [4-14](#)
 reviewing validation errors, [3-3](#)

select lists, [4-18](#)

SQL Commands, [3-7](#)

T

templates

 buttons, [4-19](#)

Theme 42

 See Universal Theme

themes

 accessibility tested, [4-2](#)
 custom, [4-1](#)
 styles, [4-2](#)
 Theme Roller, [4-2](#)
 Universal Theme, [4-1](#)

U

Universal Theme, [4-1](#), [4-6](#), [4-10](#), [4-17](#)

UT

 See Universal Theme

V

validation errors

 fixing, [3-3](#)
 rendering, [4-5](#)
 reviewing, [3-3](#)

VPAT

 See ACR

W

WAI-ARIA support, [3-4](#)

Web Content Accessibility Guidelines (WCAG),
 [2-1](#)