

# A Guide to Accessible Content for Oracle Business Intelligence Suite 11g

An Oracle White Paper  
Version 3 - December 2012  
(Updates for OBIEE v11.1.1.6.7 and higher)



## Table of Contents

Introduction .....	1
For Those New to the Issue of Accessibility.....	2
Starting Resources .....	2
New Accessibility Features in OBIEE 11g .....	3
Multiple Methods of Operation .....	5
Impact on Ad-Hoc Query Users .....	5
Impact on Designing Content for Accessibility .....	5
Impact on Assessing Accessibility .....	5
For End Users .....	6
Navigation Areas within Oracle Dashboards.....	6
Menu Area .....	6
Dashboard Header .....	7
Content Area .....	7
Selecting the Right Mode of Operation .....	7
Keyboard Navigation .....	8
Interactive Dashboards.....	8
Keyboard Navigation within a Table or Pivot .....	10
Navigating Menus and Outlines .....	10
BI Composer Wizard .....	11
Notes for Screen Reading Software.....	11
Keyboard Navigation within a Table or Pivot (Screen Reader Mode)..	11
Turning on Accessibility Mode for the UI.....	12

Using the User Preferences Dialog .....	12
Using the Login Screen .....	13
Selecting BI Composer for Sighted Keyboard-Only Users .....	14
Accessible Creation of Analytical Displays with BI Composer.....	14
Accessible Viewing of BI Publisher Reports.....	15
Accessible Creation of BI Publisher Reports.....	15
Common “Issues” with Accessibility in OBIEE .....	17
Hidden Section Headers re-appear in Accessibility Mode .....	19
Accessibility Mode works differently from “regular” mode .....	19
The “Browse/Manage...” area disappears in Accessibility Mode..	19
Hierarchy navigation plus signs disappear in Accessibility Mode .	20
Hierarchy Navigation in OBIEE v11.1.1.6.0 through v11.1.1.6.6 .....	20
Hierarchy Navigation in OBIEE v11.1.1.6.7 and higher .....	24
Action Links with 1 action don’t auto-launch in Accessibility Mode	25
I’m a sighted keyboard user. Accessibility Mode changes keys ...	25
My client wants an accessible Map display of the data .....	25
For Developers.....	27
Common Misconceptions .....	27
General Guidelines for Creating Accessible Content .....	27
Font Selection.....	28
Use of Color.....	28
Color Contrast .....	28
Accessible Charts.....	29
Dashboard-Specific Guidelines.....	30

Consistency .....	30
Simplicity .....	30
On-Screen Content.....	30
Provide sufficient Textual Information .....	30
Consistency of Graphical Elements .....	31
Alternate Displays.....	31
General Prohibitions .....	31
Specific OBIEE Techniques for Implementing Accessibility .....	31
Oracle Dashboards: Screen Layout Recommendations.....	32
Screen Organization.....	32
Navigation Targets .....	32
Drilling Control .....	35
Working with Oracle Dashboard Styles.....	38
Other Customizations .....	41
Oracle Dashboards: Dashboard Prompt Requirements .....	41
All Dashboard Prompts.....	41
Choices Lists .....	41
Calendar Widgets .....	42
Use only 1 Prompt Object per Dashboard Page .....	43
Analytical Displays: Suggested Changes from Default Settings...	43
Using View Selectors to Switch Content Displays .....	43
Using Dashboard Prompts to Switch Content Displays .....	46
Using a System Variable to Switch Content Displays .....	53
Special considerations for Exalytics deployments .....	60

Use Static Text objects to provide custom HTML Markup .....	61
Define Descriptions for all Reports .....	62
BI Publisher: Turning on Accessibility Mode for Output.....	63
BI Publisher: Creating Accessible Content Templates .....	66
Additional Enhancements .....	67
Custom On-Line Help for Accessibility Users.....	67
Links to Required Plug-Ins .....	81
Controlling Visibility of Custom Links .....	82
Conclusion .....	83

## Table of Figures

Figure 1 – The different regions of an Oracle Dashboard screen .....	6
Figure 2 – OBIEE User Preferences.....	13
Figure 3 – BI Composer Wizard.....	14
Figure 4 – BI Publisher Report View menu.....	15
Figure 5 – BI Publisher Template Builder in MS Word .....	16
Figure 6 – Installing the BI Publisher Template Builder.....	17
Figure 7 – Using "Open" in the Menu Bar instead of “Browse/Manage”.....	19
Figure 8 – Hierarchy Column in "Regular" Mode.....	20
Figure 9 – Hierarchy Column in Accessibility Mode (with Context Menu icons) .....	21
Figure 10 – Using the Context Menu Icon in Accessibility Mode.....	21
Figure 11 – Selecting Hierarchy Column Drill Targets .....	22
Figure 12 – A Drilled-Into Hierarchy in Accessibility Mode.....	22
Figure 13 – Standard Column Context Menu in Accessibility Mode.....	23
Figure 14 – Context Drilling in Accessibility Mode (through v11.1.1.6.6).....	23
Figure 15 – Mixed Column Interactions (Regular Mode) .....	24
Figure 16 – Mixed Column Interactions (Accessibility Mode in v11.1.1.6.7+) .....	24
Figure 17 – Context menus always have multiple options in Accessibility Mode .....	25
Figure 18 – Default display of a multi-section dashboard .....	33
Figure 19 – Changing a Section’s properties .....	34
Figure 20 – Dashboard with defined and visible Section Names .....	34
Figure 21 – A multi-section dashboard with section names defined.....	35
Figure 22 – Default Drill Behavior in a Dashboard.....	36
Figure 23 – Turning On the Drill-in-Place feature .....	37
Figure 24 – The Context-Preserving Effects of In-Place Drilling.....	37
Figure 25 – Oracle 11g Styles and Skins folder structure.....	39
Figure 26 – Accessing the dashboard properties.....	40
Figure 27 – Applying a Custom Style to a Dashboard .....	40
Figure 28 – Choices List: Turning off manual values entry .....	42
Figure 29 – Non-Accessible Date Between prompt.....	42
Figure 30 – Accessible Start Date prompt.....	42
Figure 31 – Multiple Apply buttons from multiple Prompt objects.....	43
Figure 32 – Single Redesigned Multi-Field Dashboard Prompt .....	43
Figure 33 – View Selector Definition.....	44
Figure 34 – View Selector in Action.....	45

Figure 35 – After selecting accessible Chart version.....	45
Figure 36 – Sample Visibility Control Dropdown .....	46
Figure 37 – Defining a Variable-based Prompt.....	46
Figure 38 – Dashboard Prompt definition using a Presentation Variable.....	47
Figure 39 – Defining the Prompt Page Settings .....	48
Figure 40 – Analysis Request as Visibility Controller .....	48
Figure 41 – Modifying an Analysis column's formula .....	49
Figure 42 – Defining a custom formula to control visibility .....	49
Figure 43 – Add a Filter based on a Presentation Variable.....	50
Figure 44 – Defining which Presentation Variable to use .....	50
Figure 45 – Dashboard Section Layout Organization .....	51
Figure 46 – Making the section's display conditional .....	52
Figure 47 – Setting the Accessible Displays section's Display Condition .....	52
Figure 48 – Switching between Standard and Accessible Displays .....	53
Figure 49 – Accessibility Mode Status: Column 1 Formula.....	54
Figure 50 – Accessibility Status Mode: Static Text Object as Status Message .....	55
Figure 51 – Accessibility Mode Status: Unfiltered display .....	56
Figure 52 – Accessibility Mode Status: Creating a Presentation Variable Filter .....	56
Figure 53 – Accessibility Mode Status: Defining the Filter Condition.....	57
Figure 54 - Accessibility Mode Status display in Regular Mode.....	57
Figure 55 - Accessibility Mode Status display in Accessibility Mode .....	57
Figure 56 – Set the view to display only the Static Text object as a status message ..	58
Figure 57 - Auto-switching Dashboard in Regular Mode .....	59
Figure 58 - Auto-switching Dashboard in Accessibility Mode.....	59
Figure 59 – The Dashboard Tools menu .....	60
Figure 60 – Turning on all APPLY Buttons .....	61
Figure 61 – Using HTML Mark-Up in a Static Text object.....	62
Figure 62 – Defining a Report's Description .....	62
Figure 63 – Accessing the Administration Screens .....	63
Figure 64 – The main OBIEE Administration screen .....	63
Figure 65 – The main BI Publisher Administration screen .....	64
Figure 66 – Turning on Accessibility Mode for BI Publisher output .....	65
Figure 67 – Including Column and Row Headers in Tables .....	66
Figure 68 – Adding a Static Text object .....	68
Figure 69 – Invoking the Static Text object's Editor.....	69
Figure 70 – Sample Accessibility Help in a Static Text object .....	69
Figure 71 – Positioning the Static Text object.....	70



Figure 72 – Final design with static In-line Help text.....	70
Figure 73 – Page-Level In-Line Help.....	71
Figure 74 – Invoking the Text object's Properties editor.....	71
Figure 75 – Defining a Page-Level Text object.....	72
Figure 76 – Dashboard Editor Multi-Column In-Line Help design .....	72
Figure 77 – Selecting the "Upload a File to the current folder" Task .....	73
Figure 78 – Link properties for custom help document .....	74
Figure 79 – Dashboard Editor layout with Link to Custom Help.....	75
Figure 80 – Link to Custom Help.....	75
Figure 81 – Link to Custom Help in Action .....	76
Figure 82 – Invoking the Section Properties Menu .....	77
Figure 83 – Hiding the Section Border .....	77
Figure 84 – Invoking the Text 1 object's Properties dialog box .....	78
Figure 85 – Final "Hidden Text" Design.....	79
Figure 86 – Hidden Text in Regular Mode .....	80
Figure 87 – Hidden Text in Accessibility Mode .....	80
Figure 88 – Defining a Link object to download a 3rd party plug-in.....	81
Figure 89 – Link to download 3rd Party Plug-in in Action .....	82

## Introduction

Oracle Business Intelligence Suite Enterprise Edition 11g (OBIEE 11g) is used by many people in different organizations. Not everyone has mobility in their hands, perfect eyesight, or other physical skills which software designers too often take for granted.

The accessibility features in OBIEE 11g aim to make aspects of navigating and using the product easier for persons with disabilities and for the aging population. The accessibility features support the use of standards-based assistive-technology hardware and software. These OBIEE 11g accessibility features fall into three general categories:

- Keyboard shortcuts that make it easier to navigate content for users with limited or no ability to use a mouse.
- Features used by third-party assistive-technology products. These features center on providing a user interface (UI) which consists of standard HTML elements that can be easily interpreted by 3<sup>rd</sup> party assistive technology products like screen readers.
- Content design capabilities that make it possible for content creators to build BI content that supports users with accessibility needs.

Remember that OBIEE is a platform for content **creation** as well as content delivery. As such, it is important to note that, while the first two accessibility feature categories are implemented “out of the box” as content **delivery** features in OBIEE, the available design **creation** capabilities are not automatic. It is the responsibility of the content designer to leverage these available features to insure that the dashboards, reports and other display objects you create do in fact adhere to accessibility standards.

## For Those New to the Issue of Accessibility

When creating content that will be consumed by a wide variety of users, providing support for users with various disabilities is not simply a good idea – it is a legal requirement in most locations throughout the world.

Discussion about accessibility across the IT industry can be found in several published books. This guide does not intend to duplicate those works. Accessibility standards and legislation exist, such as the World Wide Web Consortium (W3C), and Section 508 of the U.S. Rehabilitation Act.

### Starting Resources

For more information regarding accessibility, please refer to the links below:

- [U.S. Section 508](http://www.section508.gov/)  
http://www.section508.gov/
- [U.S. Section 508 Refresh – 2011 Draft](http://www.access-board.gov/sec508/refresh/draft-rule.htm)  
http://www.access-board.gov/sec508/refresh/draft-rule.htm
- [Web Content Accessibility Guidelines 1.0 \(WCAG 1.0\)](http://www.w3.org/TR/WCAG10/)  
http://www.w3.org/TR/WCAG10/ Starting Resources
- [Web Content Accessibility Guidelines 2.0 \(WCAG 2.0\)](http://www.w3.org/TR/WCAG20/)  
http://www.w3.org/TR/WCAG20/
- [WAI-ARIA \(The Accessible Rich Internet Applications Suite\) Overview](http://www.w3.org/WAI/intro/aria.php)  
http://www.w3.org/WAI/intro/aria.php
- [WAI-ARIA Primer](http://www.w3.org/TR/wai-aria-primer/)  
http://www.w3.org/TR/wai-aria-primer/
- [Roadmap for Accessible Rich Internet Applications \(WAI-ARIA\)](http://www.w3.org/TR/aria-roadmap/)  
http://www.w3.org/TR/aria-roadmap/

## New Accessibility Features in OBIEE 11g

OBIEE 11g has significant advances in accessibility versus OBIEE 10g that can make creation of accessible content easier. OBIEE 11g also implements accessibility mode differently, with significantly more end user features and less potential impact on performance, than does OBIEE 10g.

Comparison of Accessibility Features:

Feature	OBIEE 10g	OBIEE 11g
<b>Accessibility Mode</b>	<ul style="list-style-type: none"> <li>• Global setting in INSTANCECFG.XML for dashboards</li> <li>• Affects ALL users</li> <li>• Other settings for BI Publisher</li> <li>• Separate setting for Hyperion Interactive Reporting</li> </ul>	<ul style="list-style-type: none"> <li>• Individual setting found in My Account User Preferences</li> <li>• Checkbox in login screen as well</li> <li>• Affects <i>only</i> users who enable it</li> <li>• Setting applies to all modules (Dashboards, BI Publisher, etc.)</li> <li>• Hyperion Interactive Reporting no longer needed for accessibility support</li> </ul>
<b>HTML Mark-up Generation</b>	<ul style="list-style-type: none"> <li>• Requires global Accessibility Mode setting = On</li> <li>• Table markup only generated when table object receives focus</li> <li>• May require JAWS users to reload the screen buffer to “see” the new mark-up</li> </ul>	<ul style="list-style-type: none"> <li>• Requires user preference or login screen Accessibility Mode setting = On</li> <li>• All mark-up generated when page is first displayed</li> </ul>
<b>Creation of Alternate Displays of Charts, Graphs, Tickers, etc. for Screen Reader users</b>	<ul style="list-style-type: none"> <li>• Content Designer must create alternate displays</li> <li>• Relies on use of View Selector to switch displays</li> <li>• Requires user action to switch</li> </ul>	<ul style="list-style-type: none"> <li>• Automatically generates Table/Pivot equivalent</li> <li>• Relies primarily on Accessibility Mode = On</li> <li>• No user action required to display default alternate displays</li> </ul>

Feature	OBIEE 10g	OBIEE 11g
<b>Analytic Capabilities <i>without</i> requiring access to Answers / Analysis Editor</b>	<ul style="list-style-type: none"> <li>• Drill</li> </ul>	<ul style="list-style-type: none"> <li>• Drill</li> <li>• Pivot</li> <li>• Move to Sections/Report Prompts</li> <li>• Keep Only/Remove Only</li> <li>• Children Of/Siblings Of</li> <li>• Selection Steps</li> <li>• Add Row/Column Totals</li> <li>• Add Running Totals</li> <li>• Include / Exclude Column</li> <li>• Launch Action Framework Action Links</li> </ul>
<b>Keyboard-only Access to functionality</b>	<ul style="list-style-type: none"> <li>• Non-logical keyboard equivalents (eg., Ctrl-Shift-F3 to select the first dashboard Page)</li> </ul>	<ul style="list-style-type: none"> <li>• Logical keyboard equivalents (eg., Ctrl-Alt-P to select the first dashboard Page)</li> <li>• New Context Menus for keyboard access to advanced functionality</li> <li>• Accessibility Mode moves certain functions to Context Menus for easier access</li> </ul>
<b>Accessible Ad-Hoc</b>	<ul style="list-style-type: none"> <li>• Requires use of a separate product</li> <li>• Tool = Hyperion Interactive Reporting</li> <li>• Configured via separate install</li> <li>• Launched via separate URL into a different UI</li> <li>• Objects created are <i>not</i> native OBIEE objects</li> <li>• Objects created can only be used in OBIEE dashboards as embedded linked objects</li> </ul>	<ul style="list-style-type: none"> <li>• Uses new integrated OBIEE module</li> <li>• Tool = Oracle BI Composer Wizard</li> <li>• Configured via User Preferences setting</li> <li>• Launched via regular Create / Modify functions</li> <li>• Objects created <i>are</i> native OBIEE objects</li> <li>• Objects created can be used just like any other Answers-created OBIEE object</li> </ul>

---

**NOTE:** The minimum release level of OBIEE 11g that supports accessibility is v11.1.1.6.0. Users of earlier releases of OBIEE 11g should upgrade to v11.1.1.6.0 or higher if accessibility support is needed.

---

## Multiple Methods of Operation

The accessibility standards described by the U.S. Access Board, the World Wide Web Consortium (W3C) and other bodies has set policy that requires “at least one method of operation” that provides “equivalent functionality” for users with accessibility needs. Alternate displays and techniques are noted as being acceptable when a specific display type cannot be made accessible itself.

In other words, the accessibility standards require that at least **one** way exists to use the information in an accessible manner, but it does **not** require that **all** ways of using the system be accessible. The standards also require “equivalent” functionality but do not require **identical** functionality.

These distinctions have a direct impact on several aspects of accessibility in OBIEE. Many areas of OBIEE functionality allow for multiple methods of operation. Data displays are not necessarily single mode displays. They provide the ability for the content designer to create multiple simultaneous ways to display the same data.

### Impact on Ad-Hoc Query Users

OBIEE 11g provides two ways to create analytical data displays – the standard Analysis Editor (formerly known as Answers) and a new module called the BI Composer Wizard. Both tools create the same kind of content objects, but only BI Composer is coded for accessibility. Therefore, any keyboard-only users who are given privileges to create their own content will need to use BI Composer instead of the Analysis Editor. This is a setting configured in the User Preferences dialog box.

### Impact on Designing Content for Accessibility

Accessibility Mode is really Screen Reader Mode, so sighted users will typically not use it. As a result, data display designs cannot always rely on Accessibility Mode to auto-generate accessible displays. Report designs may need to display the same data in multiple ways. For example, a report that contains both a chart and a table will be accessible, even if the chart itself is not. The table object of the same data provides the needed “at least one method of operation” that accessibility standards require.

### Impact on Assessing Accessibility

Accessibility testers need to test the correct modules (BI Composer instead of the default Analysis Editor, for example), to use the correct mode (Accessibility Mode turned ON), and to understand the data context of the displays (does this non-accessible content **also** appear in an accessible way on the same screen). Otherwise, they will report false accessibility failures.

## For End Users

The information in this section is targeted for end users of the content delivered in an OBIEE 11g installation. It provides details about keyboard navigation and the proper methods for turning on Accessibility Mode in OBIEE 11g.

### Navigation Areas within Oracle Dashboards

To properly understand how the keyboard navigation shortcuts work within an Oracle Dashboards screen, you first need to understand how the screen is organized.

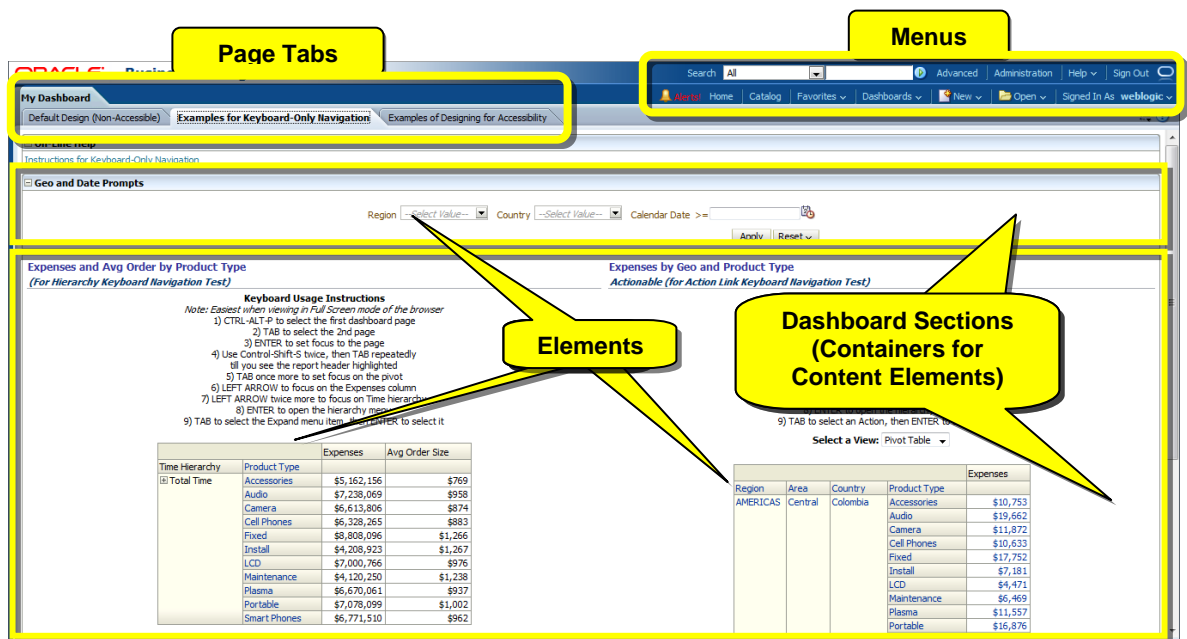


Figure 1 – The different regions of an Oracle Dashboard screen

#### Menu Area

At the top of the screen, on the right-hand side, are a series of menus that are associated with the major areas of functionality that are available to the user. Each user may see a different set of menus, depending on their role and security settings.

Items in the menu area allow you to display other non-dashboard OBIEE pages, to select functionality associated with other OBIEE modules, to create new data displays, to save dashboard filter selections as a named set for later use, to select a saved filter set to be applied, to edit the current dashboard (if your security settings allow this), to modify your user preferences, and to log out of the system.

One of the menus is the Dashboards menu. As with the menu system itself, different users may see a different list of available dashboards. The specific dashboard list a user sees is controlled by their role and security settings. Typically, the first dashboard listed for every user is the user’s personal dashboard named “My Dashboard”.

### Dashboard Header

The area immediately below the menu region is the dashboard header region. If the dashboard has more than one page, a set of named tabs will appear starting on the left side of the dashboard header. Each tab corresponds to a page in the current dashboard.

### Content Area

Below the dashboard header region is the content region. In this region will be one or more dashboard sections. A dashboard section is simply a content container. They allow a dashboard designer to group related content elements together. Sections can be used to organize content by stacking sections vertically or organizing them into columns. Sections are also used for rapid keyboard navigation to a specific portion of the dashboard content. Sections can optionally be collapsed to temporarily hide their content.

Within each dashboard section, you will have one or more content elements. These elements can be dashboard filter prompts, charts, tables or other reporting objects.

## Selecting the Right Mode of Operation

OBIEE provides multiple methods of operation for most functionality. To insure that use of OBIEE supports the end user’s accessibility needs, each user must select the mode of operation that is appropriate for their requirements. The table below documents the correct mode of operation to insure accessible functionality of the various OBIEE modules.

Desired Action	Proper Mode for Accessible Operation
Use Keyboard-Only Navigation	Works in all modes of operation
Use a Screen Reader	Turn Accessibility Mode ON
Create / Modify an Ad-Hoc Analysis	Use the BI Composer Wizard
View a BI Publisher Report	Open the report in HTML mode
Create / Modify a BI Publisher Report Design	Use the BI Publisher Template Builder for MS Word
Browse the contents of the Web Catalog	Use the Open menu in the Menu Bar (instead of the Browse/Manage area on the Home page)

How to set up or invoke each mode of operation listed will be covered later in this End User section.



## Keyboard Navigation

The available keyboard commands for Oracle Dashboards, BI Publisher and the overall Oracle BI environment are listed here in this document, grouped based on module, menu and control.

### Interactive Dashboards

The keyboard shortcuts below can be used to navigate the dashboard regions and content.

Desired Action	Keyboard Shortcut
Navigate to the Global Header (for example, to select the Skip to Content link)	Ctrl+Alt+G
Open the Dashboards menu	Ctrl+Alt+D, then ENTER
Navigate within the Dashboards menu	Up / Down Arrow keys
Open / Close the currently focused Dashboards folder	Enter
Select the Dashboard whose link has the current focus	Enter
Focus on the 1 <sup>st</sup> dashboard page tab in current dashboard (applies to multi-page dashboards only)	Ctrl+Alt+P
Navigate to the next Dashboard page tab	Tab (when focused on a tab)
Navigate to the previous Dashboard page tab	Shift+Tab
Select the dashboard page whose tab currently has focus	Enter
Move forward through sections on dashboard page	Ctrl+Shift+S
Move backward through sections on dashboard page	Ctrl+Shift+U
Move forward through elements of a section	Tab
Move backward through elements of section	Shift+Tab

### Dashboard Prompts

Prompt Control	Desired Action	Keyboard Shortcut
Dropdown Prompt (aka Choices List)	Invoke the prompt	Alt+Down Arrow key
	Navigate the prompt	Up / Down Arrow keys
	Select/Deselect one or more Values *	Spacebar
	Invoke the Search dialog	End, then Enter

Prompt Control	Desired Action	Keyboard Shortcut
Prompt Search Dialog	Select the Search Criteria (Begins, etc.)	Up / Down Arrow keys
	Execute any Search Dialog button	Spacebar or Enter
	Navigate the Found/Selected Items lists	Up / Down Arrow keys
	Select/Deselect a Found Items or Selected Items value	Spacebar
	Select multiple Found Items or Selected Items values *	Shift+Spacebar on all items after the first one
	Add /Remove a highlighted Value to/from the Selected Items List	Navigate to the appropriate Move action, then Spacebar
Date Prompt Calendar Widget	Invoke the widget	Enter or Spacebar
	Navigate within the widget	Tab or Shift+Tab
	Select a Month or Year	Up / Down Arrow keys
	Select a Day	Arrow keys when focus in the days display, then Spacebar
Checkboxes	Navigate within Checkbox list	Tab or Shift+Tab
	Select/Deselect Checkboxes *	Spacebar
	Set focus to Checkbox collection after selecting a value	Tab
Radio Buttons	Navigate within Radio Button list	Up / Down Arrow keys
	Select a Radio Button	Spacebar
	Set focus to Radio Button collection after selecting a value	Tab
List Box	Navigate within List Box items	Tab or Shift+Tab
	Select /Deselect a List Box Item	Spacebar
	Select multiple List Box items *	Ctrl+Spacebar on all items after the first one
	Set focus to List Box collection after selecting a value	Tab
Apply Button	To apply the selected filter values	Enter

Prompt Control	Desired Action	Keyboard Shortcut
Reset Button	Open Reset Button Menu	Enter
	Navigate the Menu	Tab or Shift+Tab
	Select a Menu Item	Enter

**\* Note:** Only prompts with the “Enable user to select multiple values” option turned on will allow users to select more than one item.

#### Keyboard Navigation within a Table or Pivot

Desired Action	Keyboard Shortcut
Invoke Table Navigation Mode	LeftArrow once the table or pivot receives focus
Move from cell to cell within the Table/Pivot	Up/Down/Left/Right Arrow keys
Open a Context Menu or Submenu	Enter
Navigate within a Context Menu	Tab or Shift+Tab
Exit a Menu or Submenu	Esc

#### Navigating Menus and Outlines

The following applies to the Oracle Home Page, Catalog Page, Dashboards and BI Publisher.

Control	Desired Action	Keyboard Shortcut
Menus	Open Menu	Enter
	Move Up/Down menu or submenu items	Up/Down Arrow keys
	Select Menu Item action	Enter
	Open Submenu	Enter
	Exit a menu or submenu	Esc
Catalog Folder Outline	Navigate the Outline	Up/Down Arrow keys
	Select a Catalog Folder	Enter
	Expand a Catalog Folder	RightArrow
	Collapse a Catalog Folder	LeftArrow
	Display the selected Catalog Folder’s contents	Spacebar

Control	Desired Action	Keyboard Shortcut
Catalog Content Pane	Navigate the Content Pane's list of items	Up/Down Arrow keys
	Display the selected item	Spacebar
	Navigate to an item's menu options	Tab or Shift+Tab
	Invoke an item's highlighted menu option	Enter

### BI Composer Wizard

Desired Action	Keyboard Shortcut
Invoke the Catalog Tree context menu	Ctrl+Alt+m
Read each Message in a Pop-Up window	Alt+DownArrow
Select the Back button	Shift+Alt+b
Select the Cancel button	Shift+Alt+c
Select the Finish (aka "last") button	Shift+Alt+l
Select the Next button	Shift+Alt+x

### Notes for Screen Reading Software

If you use a screen reader program, your keyboard navigation within a table object will be controlled mainly by your assistive technology. The specific OBIEE keyboard shortcuts to use (in addition to your screen reader's table navigation keys) are listed below.

---

**Note:** If you are running a version of OBIEE 11g older than the v11.1.1.6.7 patch release, you must tab in to the table rather than using any screen reader hotkeys to land on the table. Once you are in the table you can use all standard screen reader table navigation hotkeys.

---

### Keyboard Navigation within a Table or Pivot (Screen Reader Mode)

Desired Action	Keyboard Shortcut
Navigate to Next / Previous cell within the Table/Pivot and read the contents of the cell	Use your screen reader's table navigation shortcuts (Ctrl+Alt+arrows for JAWS)
Navigate to Next / Previous row and read the contents of the entire row	Use your screen reader's table navigation shortcuts

Desired Action	Keyboard Shortcut
Open a Column Header Column Menu (User must <i>first</i> navigate to the column header cell)	Enter or Spacebar
Open a Submenu	Enter or Spacebar
Navigate within a Context Menu or Submenu	Tab or Shift+Tab
Exit a Menu or Submenu	Esc

---

**Note:** Starting in v11.1.1.6.7, Accessibility Mode surfaces the default cell interaction as a link within the table markup. Users can select the default action simply by hitting Enter or Spacebar.

---

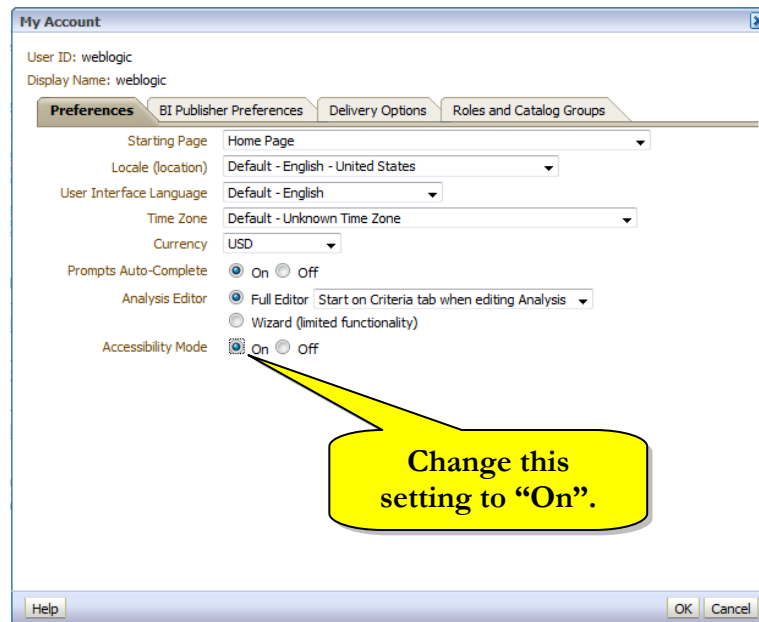
## Turning on Accessibility Mode for the UI

The OBIEE Accessibility Mode is specifically designed for users that require the use of a screen reader such as JAWS. The function of Accessibility Mode is to generate the extra HTML Markup language that is required by screen readers to be able to read the definition of the on-screen objects displayed by OBIEE. Turning on Accessibility Mode will also auto-select BI Composer as the Analytics editor tool.

### Using the User Preferences Dialog

The Accessibility Mode for the OBIEE user interface can be set for all sessions of each individual user within their User Preferences dialog. To turn on this Mode (in a default, non-customized configuration):

1. Log into the Oracle Business Intelligence web page.
2. Hit Ctrl+Alt+D to enter the menu area. The “Dashboard” menu will be selected.
3. Hit Tab three times to set focus to the “Signed in as...” User Menu.
4. Hit Enter to invoke the User Menu.
5. Tab to the “My Account” menu item and hit Enter. The following User Preferences dialog box will be displayed:



**Figure 2 – OBIEE User Preferences**

6. Hit Tab repeatedly to set focus to the Accessibility Mode “Off” Radio Button, then use the LeftArrow key to select the “On” Radio Button.

Alternately, you can invoke the screen reader’s “List Radio Buttons” hotkey (Ctrl+Insert+R for JAWS) to directly navigate to the Accessibility Mode “On” Radio Button and hit the Spacebar to activate it.

7. Tab to the OK button to set the selection and exit the dialog box.

---

**Note:** When setting this mode from the user’s Preferences screen, you must refresh the screen display after exiting the dialog before you will see the effects of the change.

---

### Using the Login Screen

An alternate method for turning on Accessibility Mode exists in the login screen. This method can be used for deployments that either are not allowed or are not able to identify accessibility-needs users ahead of time.

When logging in, the user can tab past the password field and the “Sign In” button to activate the “Accessibility Mode” checkbox.

---

**Note:** This will turn on Accessibility Mode for the user’s current session only. If the User Preferences screen is set to activate Accessibility Mode, the login screen checkbox does not need to be used.

---

## Selecting BI Composer for Sighted Keyboard-Only Users

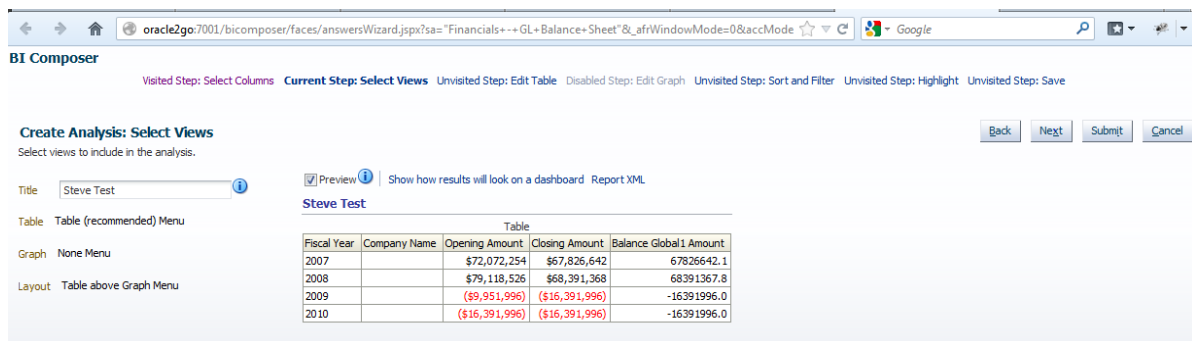
Sighted users with accessibility needs may not want to use Accessibility Mode, with its auto-conversion of graphical displays to tabular ones. In these cases, users can select to use BI Composer instead of the default Analysis Editor without turning on Accessibility Mode. Simply select the “Wizard” option as the Analysis Editor in the User Preferences dialog shown above in Figure 2.

## Accessible Creation of Analytical Displays with BI Composer

OBIEE provides a specific UI which supports creation of Analytical Displays by users with accessibility needs. This tool is called BI Composer. BI Composer is a simple-to-use multi-step wizard that steps a user through the process of creating Analytical Displays. BI Composer allows you to quickly and easily create, edit, or view analyses without the complexities of the default Analysis editor.

The main components of the BI Composer wizard are as follows:

- At the top of the wizard is the BI Composer process steps train, which contains buttons for each step in the wizard.
- On the left side is the Catalog Tree area. This is where the user will select the items to include in the analysis.
- To the right of the Catalog Tree area is the panel area, where the selected components for each step are displayed.
- To the far right are the **Back**, **Next**, **Finish**, and **Cancel** navigation buttons.



The screenshot shows the BI Composer wizard in the 'Select Views' step. The progress bar at the top indicates the current step. The main area shows a title field with 'Steve Test' and a preview table. The table has the following data:

Fiscal Year	Company Name	Opening Amount	Closing Amount	Balance Global1 Amount
2007		\$72,072,254	\$67,826,642	67826642.1
2008		\$79,118,526	\$68,391,368	68391367.8
2009		(\$9,951,996)	(\$16,391,996)	-16391996.0
2010		(\$16,391,996)	(\$16,391,996)	-16391996.0

**Figure 3 – BI Composer Wizard**

BI Composer creates the same type of analytical report objects as does the main Analysis Editor. Users of BI Composer can create a query from scratch, define a table, pivot and/or chart object in which to display the data and can set formatting parameters, sort and filter settings, and data highlighting for the display. For more complete information on the use of BI Composer, please refer to Chapter 14 in the OBIEE User’s Guide.

## Accessible Viewing of BI Publisher Reports

The BI Publisher module supports the display of reports in a wide array of formats, all invoked from the same Report object. Users with accessibility needs must select only the HTML display mode to support accessible viewing of the report.

If a BI Publisher report is not set to display in HTML mode as its default, the user can self-select the HTML display mode. Once a report has been selected and displayed, Tab to the “Report View” menu, then hit Enter to invoke the menu. Use the Up and Down Arrow keys to navigate within the menu. Select the HTML display mode and hit Enter to invoke it.

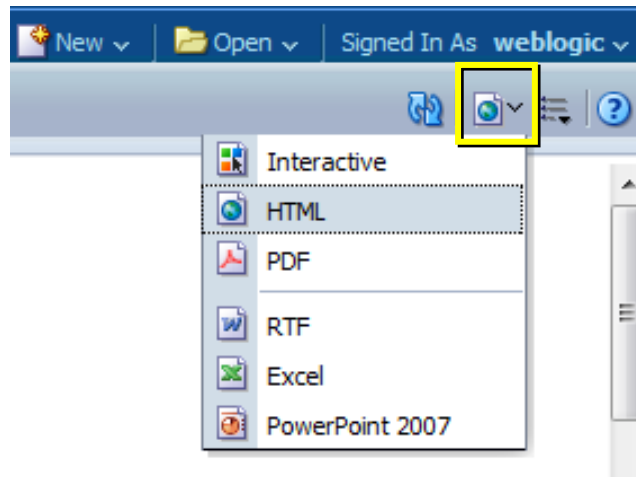


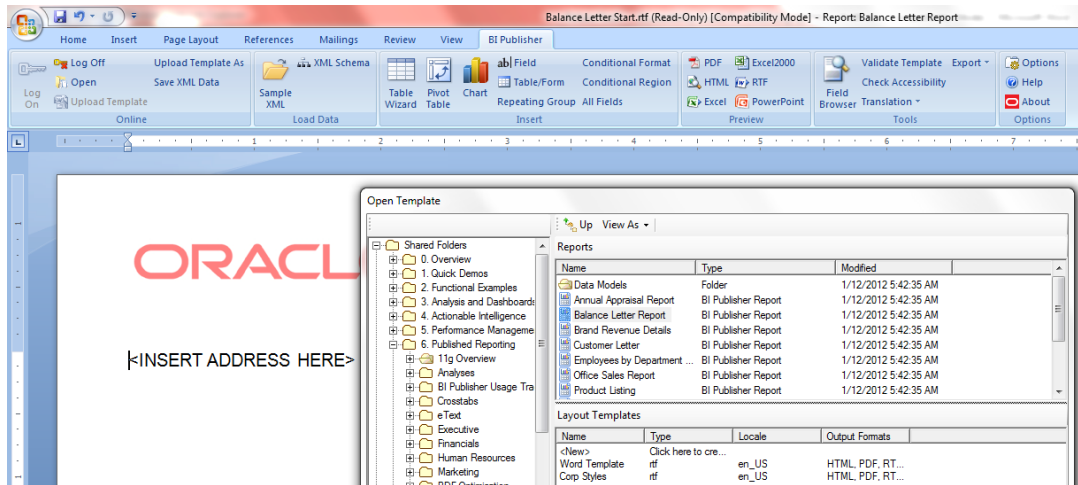
Figure 4 – BI Publisher Report View menu

## Accessible Creation of BI Publisher Reports

The BI Publisher module of OBIEE has two main methods of creating and modifying BI Publisher report designs – an on-line visual Layout Editor, and a desktop Template Builder add-in for Microsoft Word. The on-line visual BI Publisher Layout Editor has not been coded for accessibility and so should not be used by end users with accessibility requirements.

The desktop Template Builder for MS Word supports accessibility needs users. It is also designed to be used by advanced content designers who wish to leverage some of the more robust BI Publisher functionality, whether or not these designers require accessibility support. The BI Publisher Template Builder is a standard Microsoft Office Add-in module that appears as an additional menu or ribbon in MS Office.





**Figure 5 – BI Publisher Template Builder in MS Word**

The BI Publisher Template Builder must first be installed into the BI Publisher report designer user’s local copy of MS Word. It is available in both 32-bit and 64-bit editions. Make sure you install the correct version for your local copy of MS Office.

Supported MS Office Version	Required BI Publisher Template Builder Edition
MS Word 2003, 2007 SP2	32-bit edition only (since these versions of MS Word are 32-bit only)
MS Word 2010	32-bit or 64-bit, depending on which edition matches the installed edition of MS Word.  Check which edition you are using from inside MS Word by using the File / Help menu.

The BI Publisher Template Builder can be downloaded and installed directly from the Get Started... area of the OBIEE Home Page (see the following screen shots).



Figure 6 – Installing the BI Publisher Template Builder

## Common “Issues” with Accessibility in OBIEE

Many questions that arise about accessibility in OBIEE are common across most installations. They often are based on an incomplete knowledge of all the operational capabilities available in the total Oracle BI platform. Remember that accessibility regulations require at least one method of equivalent operation that is accessible. However, accessibility does not require that *all* methods of accomplishing a task must be accessible. If multiple methods exist, only one of them needs to be accessible.

This means that questions often get generated by users who only know a non-accessible method of operation and who don't realize there may be another way to accomplish the same function in a manner that *is* accessible.

Specifically in an OBIEE context, testers must insure they are using the correct mode of operation that supports accessibility. These are summarized in the following table:

Operation	Desired Action	Proper Method for Accessible Operation
Consumption	Use Keyboard-Only Navigation	Works in all modes of operation
	Use a Screen Reader	Turn Accessibility Mode ON
	View a BI Publisher Report	Open the report in HTML mode
	Browse the contents of the Web Catalog	Use the Open menu in the Menu Bar (instead of the Browse/Manage area on the Home page)
Creation	Create / Modify an Ad-Hoc Analysis	Use the BI Composer Wizard
	Create / Modify a BI Publisher Report Design	Use the BI Publisher Template Builder for MS Word

Other questions may arise because a content developer wants to make a display operate in a certain manner which violates one or more accessibility requirements. Turning on Accessibility Mode will alter the way OBIEE displays screens, and this may not be to a particular person's liking. However, the settings and operational characteristics of OBIEE are designed to enforce the current accessibility requirements noted in Section 508 of the U.S. Rehabilitation Act and in the WCAG guidance. In these cases, the content designer must modify their particular design preferences in favor of an accessible design alternative. The developer section of this whitepaper provides design guidelines and work-arounds that enable accessible content design with minimal impact on the content designer.

Content designers must remember two important concepts:

- 1) Accessible design is not just a “nice to have” feature. It is a **legal** requirement in all public sector organizations world-wide, as well as in most commercial settings.
- 2) The advantage of the requirement of “at least one method of operation that is accessible” can be applied to content design, too. This means a content developer can continue to use non-accessible designs **as long as** the same screen includes an accessible display of the same data. In other words, with careful co-displayed alternative designs, a content developer can “have their cake and eat it too”.

As a set of examples, here are a few situations that tend to come up in nearly every implementation of OBIEE that requires accessibility, along with the appropriate response:

## Hidden Section Headers re-appear in Accessibility Mode

Accessibility regulations require that sufficient information be made available for screen navigation by users with disabilities. In an OBIEE context, this means that section headers are required so that accessibility users have a “landing site” for hot-key navigation. As a result, Accessibility Mode forces all section headers to be visible. This is a built-in feature of Accessibility Mode, so as to insure the system follows accessibility guidelines.

The Navigation Targets topic in the “For Developers” section of this whitepaper outlines how to leverage this feature to your best advantage to facilitate rapid screen navigation for accessibility-needs users. This will also be an advantage to “touch-typist” users who want better keyboard-based navigation.

## Accessibility Mode works differently from “regular” mode

Accessibility Mode is explicitly designed to operate differently so as to tailor the user interface to operate in a fashion that provides support for users of screen readers, and for keyboard-only access. Sighted users who are testing for accessibility need to understand the requirements and functionality of assistive technologies like screen readers and how a non-sighted user would interact with the system. Testers also need to be familiar with all methods and aspects of OBIEE operation. Remember that, for a sighted user, their favorite way of interacting with OBIEE may not be the accessible method required by law. Therefore, testers need to be familiar with what Accessibility Mode does to change the UI to support accessibility-needs users and then test using the method of operation that has been coded for accessibility.

## The “Browse/Manage...” area disappears in Accessibility Mode

This may sometimes be stated as “When I turn on Accessibility Mode, I can’t browse the Web Catalog.” This issue gets mentioned because users don’t realize that there are actually three different ways to accomplish the same task, only one of which is accessible.

So when Accessibility Mode is turned on, the screen displays remove all non-accessible methods of operation. In the case of browsing the Web Catalog, this means the “Browse/Manage...” area of the OBIEE Home page, and the “Catalog” menu item disappear.

Users with accessibility needs must be informed that they need to use the “Open” menu to browse the Web Catalog instead of these other two methods.

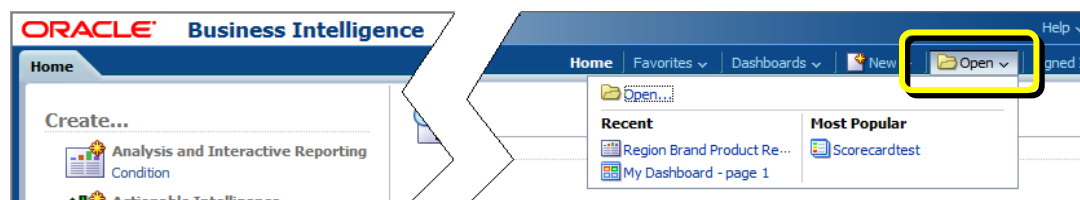


Figure 7 – Using "Open" in the Menu Bar instead of “Browse/Manage”

## Hierarchy navigation plus signs disappear in Accessibility Mode

One requirement of accessibility regulations is that a mouse must not be required to invoke functionality. As a result, some of the standard methods of accessing functionality in OBIEE need to change when Accessibility Mode is turned on. Navigating hierarchical columns is one of the areas affected. When Accessibility Mode is activated, many mouse-oriented functions are moved to a new accessibility-mode-only context menu icon located to the right of the row or column header text. This icon can be landed on and invoked via regular keyboard-only navigation. Users of OBIEE in Accessibility Mode need to become familiar with all the functionality that is surfaced to them through this special context menu icon.

This is what happens with navigation of hierarchy columns in Accessibility Mode. The ability to drill into a hierarchy moves to the “Keep Only” menu item of the column header’s context menu icon. You will find other areas of functionality contained in that menu icon, as shown below.

### My Report (Alternate Views)

		AMERICAS		
		Revenue	Avg Order Size	Avg Order Size Alert
Time Hierarchy	Product Type			
 Total Time	Accessories	\$2,420,987	778	
	Audio	\$3,337,789	964	
	Camera	\$3,102,766	878	
	Cell Phones	\$2,858,080	874	
	Fixed	\$3,999,345	1,262 	Over Limit
	Install	\$1,919,428	1,249 	Over Limit
	LCD	\$3,239,343	969	
	Maintenance	\$1,925,632	1,253 	Over Limit
	Plasma	\$3,051,045	925	
	Portable	\$3,362,291	1,011 	Over Limit
	Smart Phones	\$3,220,883	982	
<b>Grand Total</b>		<b>\$32,437,589</b>	<b>987</b>	

Figure 8 – Hierarchy Column in "Regular" Mode

### Hierarchy Navigation in OBIEE v11.1.1.6.0 through v11.1.1.6.6

In releases prior to the v11.1.1.6.7 edition, hierarchy navigation is performed using the column header menu, as shown in the following figures.

**My Report (Alternate Views)**

		My Report (Al		
		AMERICAS	AMERICAS	AMERICAS
		Revenue	Avg Order Size	Avg Order Size Alert
Time Hierarchy	Product Type			
Total Time	Accessories	\$2,420,987	778	
Total Time	Audio	\$3,337,789	964	
Total Time	Camera	\$3,102,766	878	
Total Time	Cell Phones	\$2,858,080	874	
Total Time	Fixed	\$3,999,345	1,262	Over Limit
Total Time	Install	\$1,919,428	1,249	Over Limit
Total Time	LCD	\$3,239,343	969	
Total Time	Maintenance	\$1,925,632	1,253	Over Limit
Total Time	Plasma	\$3,051,045	925	
Total Time	Portable	\$3,362,291	1,011	Over Limit
Total Time	Smart Phones	\$3,220,883	982	
<b>Grand Total</b>		<b>\$32,437,589</b>	<b>987</b>	

**Figure 9 – Hierarchy Column in Accessibility Mode (with Context Menu icons)**

To navigate the Time hierarchy in Figure 9, a user would use the keyboard to navigate to the pivot table, then TAB to set the focus on the Time Hierarchy context menu icon. Hitting ENTER will invoke the following menu.

**My Report (Alternate Views)**

		My Report (Alternate Views) Pivot Table				
		AMERICAS	AMERICAS	AMERICAS	APAC	APAC
		Revenue	Avg Order Size	Avg Order Size Alert	Revenue	Avg Order Size
Time Hierarchy	Product Type					
Total Time					\$1,139,582	766
Total Time					\$1,584,765	947
Total Time					\$1,407,890	855
Total Time					\$1,464,702	803

Collapse all items for this column	3					
Collapse all items in view	4					
Sort Column	3					
Keep Only	4					
Remove						
Show Row level Grand Total						
Show Column level Grand Total						
Exclude column						
Include column						
Hide Column						
Move Column						

Top X						
Bottom X						
Members where Revenue is greater than						
Members where Revenue is less than						
Members where Avg Order Size is great...						
Members where Avg Order Size is less t...						
Members where Avg Order Size Alert is...						
Members where Avg Order Size Alert is...						
Members At Hierarchy Levels...						

**Figure 10 – Using the Context Menu Icon in Accessibility Mode**

The actual drilling into the hierarchy is done via the “Members at Hierarchy Levels...” sub-menu. This will invoke a dialog box that allows the user to select the drill targets via keyboard accessible checkboxes. Simply use the Tab and Shift-Tab keys to navigate and the Spacebar to turn a checkbox on or off.

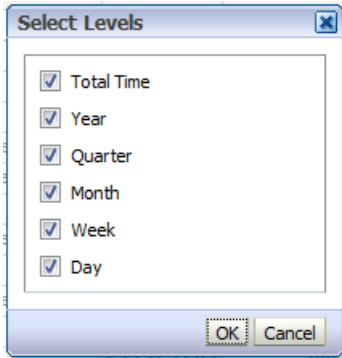


Figure 11 – Selecting Hierarchy Column Drill Targets

Selecting all the available checkboxes illustrated in Figure 11 produces the display shown below.

**My Report (Alternate Views)**

		My Re		
		AMERICAS	AMERICAS	AMERICAS
		Revenue	Avg Order Size	Avg Order Si
Time Hierarchy	Product Type			
Total Time	Accessories	\$2,420,987	778	
Total Time	Audio	\$3,337,789	964	
Total Time	Camera	\$3,102,766	878	
Total Time	Cell Phones	\$2,858,080	874	
Total Time	Fixed	\$3,999,345	1,262	Over Li
Total Time	Install	\$1,919,428	1,249	Over Li
Total Time	LCD	\$3,239,343	969	
Total Time	Maintenance	\$1,925,632	1,253	Over Li
Total Time	Plasma	\$3,051,045	925	
Total Time	Portable	\$3,362,291	1,011	Over Li
Total Time	Smart Phones	\$3,220,883	982	
2008	Accessories	\$702,506	690	
2008	Audio	\$1,136,935	989	
2008	Camera	\$1,184,235	1,004	Over Li
2008	Cell Phones	\$946,920	846	
2008	Fixed	\$1,203,400	1,175	Over Li
2008	Install	\$740,875	1,433	Over Li
2008	LCD	\$1,129,248	1,028	Over Li
2008	Maintenance	\$743,156	1,480	Over Li
2008	Plasma	\$1,066,248	961	
2008	Portable	\$941,145	885	
2008	Smart Phones	\$1,073,244	1,010	Over Li
2008 Q1	Accessories	\$94,318	385	
2008 Q1	Audio	\$170,254	655	
2008 Q1	Camera	\$175,183	698	

Figure 12 – A Drilled-Into Hierarchy in Accessibility Mode

On non-hierarchical columns, the column header context menu icon can be used to invoke the default drill action.

My Report			
	AMERICAS	AMERICAS	AMERICAS
	Revenue	Avg Order Size	Avg Order Size A
Product Type			
Accessories			
Audio			
Camera			
Cell Phones			
Fixed			
Install			
LCD			
Maintenance			
Plasma			
Portable			
Smart Phones			
Accessories			
Audio			
Camera			

Figure 13 – Standard Column Context Menu in Accessibility Mode

Targeted drilling is also available. If a screen reader user navigates into the data cells of the table using their screen reader’s hotkeys, they can hit the spacebar to select the current row, then Shift+Enter to invoke the cell-specific context menu.

In the example shown below in Figure 14, the Accessibility Mode user has navigated to row 5, column 2 of the table and hit the spacebar to select the Total Time Camera row. Hitting Shift+Enter followed by Enter selects the default Drill action and results in the focused drill shown below.

Hierarchy Keyboard Nav Test Pivot Table			
Time Hierarchy	Product Type	Expenses	Avg Order Size
Total Time	Accessories	\$5,161,698	\$769
Total Time	Audio	\$7,237,362	\$958
Total Time	Camera	\$6,614,105	\$874
Total Time	Cell Phones	\$883	
Total Time	Fixed	\$1,266	
Total Time	Install	\$1,267	
Total Time	LCD	\$976	
Total Time	Maintenance	\$1,238	
Total Time	Plasma	\$937	
Total Time	Portable	\$1,002	
Total Time	Smart Phones	\$963	

Hierarchy Keyboard Nav Test Pivot Table				
Time Hierarchy	Product Type	P1 Product	Expenses	Avg Order Size
Total Time	Camera	7 Megapixel Digital Camera	\$3,489,231	\$922
Total Time	Camera	MPEG4 Camcorder	\$3,124,873	\$825

P2 Product Type is equal to / is in Camera

[Return](#)

Figure 14 – Context Drilling in Accessibility Mode (through v11.1.1.6.6)



### Hierarchy Navigation in OBIEE v11.1.1.6.7 and higher

Starting with the v11.1.1.6.7 release, Accessibility Mode adds the default interaction to the HTML markup of each dimension cell in a table or pivot. A pivot with a mix of hierarchical columns and regular columns, and one column with Action Links defined would look like this in regular mode:

#### Global Sales Revenue Report

				Revenue
Offices Hierarchy	Brand	Line of Business	Product Type	
☒ All Offices	BizTech	Communication	Cell Phones	\$667,085.24
			Smart Phones	\$636,392.75
		Electronics	Accessories	\$367,652.40
			Audio	\$828,869.61
	FunPod	Digital	Camera	\$615,751.92
		Games	Fixed	\$404,391.02
			Portable	\$479,857.06
	HomeView	Services	Install	\$73,006.50
			Maintenance	\$98,786.94
		TV	LCD	\$393,941.84
			Plasma	\$434,264.72

**Figure 15 – Mixed Column Interactions (Regular Mode)**

Notice that, for a sighted user, they know an interaction exists by the presence of either an expand/collapse icon (for hierarchical columns) or the data cell value being displayed in blue text and changing the cursor shape when the mouse hovers over the text.

When switching to Accessibility Mode, starting in v11.1.1.6.7 OBIEE will display the pivot as shown here:

#### Global Sales Revenue Report

Global Sales Revenue Report Pivot Table				
Offices Hierarchy	Brand	Line of Business	Product Type	Revenue
All Offices (Expand)	BizTech (Drill)	Communication (Action links)	Cell Phones (Drill)	\$667,085.24
All Offices (Expand)	BizTech (Drill)	Communication (Action links)	Smart Phones (Drill)	\$636,392.75
All Offices (Expand)	BizTech (Drill)	Electronics (Action links)	Accessories (Drill)	\$367,652.40
All Offices (Expand)	BizTech (Drill)	Electronics (Action links)	Audio (Drill)	\$828,869.61
All Offices (Expand)	FunPod (Drill)	Digital (Action links)	Camera (Drill)	\$615,751.92
All Offices (Expand)	FunPod (Drill)	Games (Action links)	Fixed (Drill)	\$404,391.02
All Offices (Expand)	FunPod (Drill)	Games (Action links)	Portable (Drill)	\$479,857.06
All Offices (Expand)	HomeView (Drill)	Services (Action links)	Install (Drill)	\$73,006.50
All Offices (Expand)	HomeView (Drill)	Services (Action links)	Maintenance (Drill)	\$98,786.94
All Offices (Expand)	HomeView (Drill)	TV (Action links)	LCD (Drill)	\$393,941.84
All Offices (Expand)	HomeView (Drill)	TV (Action links)	Plasma (Drill)	\$434,264.72

**Figure 16 – Mixed Column Interactions (Accessibility Mode in v11.1.1.6.7+)**

Notice how the table markup now displays each column's default interaction as part of the data value in the cell. Since this is integrated into the table markup of the display, screen reader users

will hear spoken to them both the value and the interaction (if one is defined). This way, screen reader users know whether an interaction exists and what that interaction will be.

### Action Links with 1 action don't auto-launch in Accessibility Mode

Default behavior for single-action Action Links is to immediately launch the action when a user clicks on a row header, column header or data cell that has an action link defined. However, when Accessibility Mode is turned on, this auto-launch behavior does not work with keyboard-based navigation. This happens because, with Accessibility Mode on, certain right-click functionality gets moved to the Action Link context menu to facilitate keyboard-only users. As a result, the Action Link menu *always* has more than one item in Accessibility Mode.

Product Type	Expenses
Accessories	\$10,753
Audio	\$19,662
Camera	\$
Cell Phones	\$
Fixed	\$
Install	\$7,181

Figure 17 – Context menus always have multiple options in Accessibility Mode

### I'm a sighted keyboard user. Accessibility Mode changes keys

OBIEE Accessibility Mode is designed primarily for screen reader users. Accessibility regulations require that OBIEE operations not interfere with the operation of 3<sup>rd</sup> party assistive technologies. So as not to interfere with normal screen reader operations, certain hotkeys change functionality when Accessibility Mode is on.

---

NOTE: Sighted keyboard users can turn Accessibility Mode off if they want. Keyboard navigation will still operate, and screens will not auto-convert visual data displays into a tabular presentation when Accessibility Mode is off.

---

### My client wants an accessible Map display of the data

Maps and other graphical displays are by nature not accessible. Users can, of course, turn on Accessibility Mode, but this will turn the Map display into a table. This is fine for screen reader users, but for sighted users with other disabilities, this may not be an ideal solution.

Content designers can take advantage of the “at least one method of operation” clause to provide visual users with a Map display *plus* a tabular display of the same data. Either a table or a pivot object can be used. Since the tabular display will be accessible to sighted accessibility-needs users, the context of the *entire* display makes this solution accessible.

---

**Developers:** Make sure you take sighted accessibility-needs users into account when designing content. See the specific topics in the “For Developers” section for complete details.

---

## For Developers

A number of general guidelines exist that should always be kept in mind when designing content that will be consumed by a variety of people with differing abilities. These concepts apply to any content you create, whether or not it is OBIEE content. In addition, there are specific OBIEE tool features you need to use to ensure your OBIEE content designs support accessibility requirements.

### Common Misconceptions

Many developers make certain assumptions about technology and accessibility. Some of the more common ones include:

- HTML automatically equals accessible
- Accessible tools automatically create accessible content
- Automated testing tools can reliably determine accessibility

None of these assumptions, however, is correct. Developers can create non-accessible content using HTML. A tool that can produce accessible content may not do so by default, or may allow a developer to select options that will turn off the accessible features within existing accessible content. Automated testing tools do not always interact with content the same way end users will. Automated testing tools also have no ability to understand context. For example, an OBIEE report may contain a non-accessible chart as well as an accessible table display of the same data. Since at least one accessible method of reaching the data appears on the report, the report in its entire context is accessible. Yet, the limitations of automated testing tools would result in erroneously flagging this report as non-accessible.

As a result, determining the level of accessibility is a complex process that should involve the use of automated tools, manual testing, good judgment, and testing with persons with disabilities. When creating content, developers must be aware of certain common practices to ensure the content is accessible to all.

### General Guidelines for Creating Accessible Content

Always remember to consider the fact that multiple disabilities exist and that multiple disabilities may manifest in the same individual. You also need to remember that there are varying degrees of certain disabilities (such as the various types of color vision deficiency). Your designs must take all these possibilities into account.

### **Font Selection**

Users with low visual acuity will often use screen magnification software to make the screen easier to read. The fonts you use should be readable even when magnified by accessibility tools by as much as 20 times. Some fonts do not display well when magnified, while others do.

Oracle Dashboards use style sheets to set standard display definitions. Make sure these style sheets consistently use font selections that magnify well. That way, content creators will automatically default to using fonts that are accessible.

### **Use of Color**

Many different types of color vision deficiency exist, from an inability to see the difference between one common color pair such as red-green (the most common deficiency), all the way to full color blindness where a person can only see varying shades of grey and black. Using just color to convey critical information means that certain users will not be fully aware of all the pertinent information about a subject. And, of course, a blind user will need any information conveyed by color to also be present in an alternate textual format.

As a developer, this means that you must not create any content that provides key information by color alone.

### **Examples of Inaccessible Designs**

One example of a non-accessible design is to denote negative numbers solely by coloring the text red. Another example is a typical “stoplight” indicator where the only context information comes from its color – green for good and red for bad.

### **Alternate Accessible Designs**

Use of color is acceptable only as long as it is accompanied by some other indication of the same information. For example, you can add a minus sign or parentheses to denote negative numbers in tables and pivots. For stoplight displays, you can add descriptive text (e.g. “Status: good”) and use different shaped icons in addition to the color (e.g., green circles for “good”, yellow triangles for “warning” and red octagons for “bad”).

Remember you need to consider multiple disabilities, so text should always be included. That way, the information will also be provided to users relying on screen readers.

### **Color Contrast**

Because color vision deficiency can also manifest as an inability to distinguish between subtle shades of similar colors, overall color design of all screen elements must provide a large amount of contrast. It is recommended that you try to achieve a minimum of a 4.5:1 color luminosity contrast ratio. (This is the contrast ratio set by the WCAG 2.0 standards.)

For example, use black text on a white background instead of dark grey text on a light grey background.

An example of a tool that can test for the proper level of contrast can be found at <http://www.paciellogroup.com/resources/contrast-analyser.html>. If you want to see what your web site looks like to individuals with various types of color vision deficiency, try the tools at the <http://colorfilter.wickline.org/> web site.

### Accessible Charts

Sighted accessibility-needs users will typically prefer to view charts as-is instead of using Accessibility Mode to auto-convert them to tables. However, the default settings for most chart types will not create accessible charts automatically. As a result, content designers need to take this requirement into account when designing chart displays.

Different types of charts require different design elements to insure that they are accessible. Nearly all software packages create pie charts, bar charts and line charts that default to using only color to distinguish between different data series. Therefore, the content designer must change the default design to use some other display means to tell which data series is which.

Recommended changes are shown in the table below:

Chart Type	Change Recommended for Accessibility
Pie	Use different cross-hatch patterns for each data series
Bar	Use different cross-hatch patterns for each data series, or switch to using a Bar-Line chart
Line	Use different data marker shapes for each data series, with markers large enough to be distinguishable from the lines and from each other

Additionally, to insure that charts are accessible, you should always include the following chart elements:

- Chart Legend
- X-Axis and Y-Axis labels
- Data Labels for each data item in a series

Finally, to promote a high color luminosity contrast ratio, charts should not use colored or patterned chart backgrounds, as this may mask the data for certain color deficient users.

## Dashboard-Specific Guidelines

A number of general guidelines apply to how you design dashboards for accessibility. These guidelines apply to any technology that creates dashboards. Most are common sense items which are simply good dashboard design concepts that will be helpful for *all* users.

### Consistency

When creating multiple dashboards, make the structure consistent. If multiple dashboards contain similar functions or content, keep those links or forms in the same place on all dashboards. Buttons and links that have the same functions or destinations should have the same text and labels. Icons used for the same function should also have the same ALT text associated with them throughout the product.

### Simplicity

Try to keep dashboards simple. Do not try to put too many objects on one screen. It's better to have multiple pages that are easy to navigate than one that is cluttered and difficult to navigate.

### On-Screen Content

Just like in charts, to promote a high color luminosity contrast ratio, do not use colored or patterned dashboard backgrounds. In addition, the styles you use should also support high contrast between background and text, both in the dashboard header area as well as in the tabs on multi-page dashboards. Finally, you should place the most important content at the top of the page so users will be able to access it without having to navigate the entire screen.

### Provide sufficient Textual Information

Graphics cannot be read by assistive technologies, and low vision users may not be able to discern the meaning of a graphical element. Therefore, all graphical elements must have additional text to describe the functionality.

Graphical icons need to have what is known as ALT text – good descriptive text associated with the graphical element that adequately describes its function when selected. If a graphical element is merely present for aesthetics and has no functional purpose, it still needs a null ALT text (alt="") assigned to it so that screen readers know it should be skipped.

Other graphical elements that do not support creation of ALT text should have text fields added to the top or side to denote functionality, such as “Select a Display View Below”.

### **Consistency of Graphical Elements**

When graphical elements are used to identify controls, status indicators, or other programmatic elements, make sure that the meaning assigned to each graphical element is consistent throughout the application.

### **Alternate Displays**

For displays that are inherently visual such as interactive GIS maps or audio-video feeds, no method may exist for making these specific content elements directly accessible. When you deploy this kind of content, you must also provide a text-based equivalent display of the same information with similar interaction capabilities. Typically this means either creating an equivalent table/pivot of the related data (if applicable), or providing a caption and text description in the case of audio-visual content.

In OBIEE 11g, this requirement is made easier when the Accessibility Mode is turned on. OBIEE will automatically convert graphical displays to tabular ones when a user turns on Accessibility Mode in their User Preferences setting. Content developers must, however, be aware that this auto-conversion only applies to content created with the native OBIEE design tools. Other “3<sup>rd</sup> party” content can be displayed in OBIEE dashboards which is sourced from other tools. The content developer must insure that these content elements also provide tabular display options for graphical content.

### **General Prohibitions**

Certain application features should not be used at all, such as elements that blink with a frequency between 2Hz and 55Hz, or that use excessive animation. Make sure that you are familiar with all legally mandated design prohibitions that apply in your locality as well and avoid using those elements.

## **Specific OBIEE Techniques for Implementing Accessibility**

In addition to the above-listed general guidelines that apply to any software development project, certain techniques that are specific to OBIEE 11g should also be employed. The full OBIEE 11g product suite consists of a number of modules, each of which covers different aspects of business intelligence functionality. Therefore, a completely accessible BI solution requires the use of several different specific modules. In addition, these different modules employ different techniques to create accessible content.

Specifically, the modules included in an OBIEE 11g license that apply for accessibility purposes are:

- Oracle Dashboards – for content consumption and casual user “ad-hoc queries” via filter selections, Column Selector and View Selector controls



- BI Composer – for ad-hoc querying, reporting and data analysis
- BI Publisher in HTML mode – for display of production (canned) reporting
- BI Publisher Template Builder for MS Word – for creation of production (canned) reporting designs

Each of these modules requires specific design techniques to produce accessible content. These specific design methodologies are discussed below.

## **Oracle Dashboards: Screen Layout Recommendations**

Aside from general guidelines, certain technical features in Oracle Dashboards must be leveraged to create accessible content, most without impacting all users. However, certain aspects of assistive technologies may require slight deviations from normal Oracle Dashboards development.

### **Screen Organization**

Normally, a dashboard designer can place content elements anywhere on the screen. The requirements of assistive technologies may force certain elements to be placed according to more formal rules.

Accessibility standards require that the order of the content on a page must make logical sense. In addition, screen readers may not always be able to recognize that screen content they have already scanned has changed since it was read. Therefore, dashboard designers need to place any controls that might change the screen's content in a location on the screen that is before the content that would be affected by the control. In other words, the affected content objects should be “downstream” of any controls that might change their data.

### **Navigation Targets**

Screen readers require sufficient textual information to tell the user the navigation context of their current location. Since Oracle Dashboards are designed by populating containers called sections, you must design each section in a way that provides the information a user needs to be able to tell where they are and where they have been. Being able to use the Next or Previous Section keyboard shortcuts only has meaning if the user can know which section they want.

By default, dashboard sections are given sequential names (“Section 1”, “Section 2”, etc.) and these default names are not surfaced to the end user. The accessibility requirement to support textual information for screen readers means that dashboard designers must create meaningful names for each section in a dashboard and make those section names “visible”. Turning on this section naming feature will generate sufficient textual annotations. These annotations will be read by screen reader assistive technology and inform users of their current location within the dashboard.

NOTE: The user’s Accessibility Mode **must** be turned ON in order for the additional HTML mark-up to be generated that will allow a screen reader to announce the Section names on the dashboard.

The following screen shot illustrates the default configuration of a dashboard, without the section name feature. Each element on the screen has an element name (“Simple Regional Table” and “List of Offices”) but the sections themselves do not.

Note that, while a screen reader user can still use keyboard shortcuts to navigate the dashboard, they will not have any information about which section they are in until they navigate to a specific element within the current section. And without the additional section information, screen reader users will not be able to leverage certain assistive technology features, such as “Show Headers List”.

The screenshot shows an Oracle Business Intelligence dashboard with two sections. The first section, titled 'Simple Regional Table', contains a table with the following data:

Region	Area	Revenue
AMERICAS	Central	495,996
	North America	28,770,757
	South America	3,170,836
APAC	East	1,533,773
	North	1,785,600
	South	9,036,360
	West	3,054,076
EMEA	Africa	2,584,860
	Eastern	1,465,304
	Europe	4,900,099
	Middle East	2,596,991
	North Africa	675,218
	Northern	9,930,130

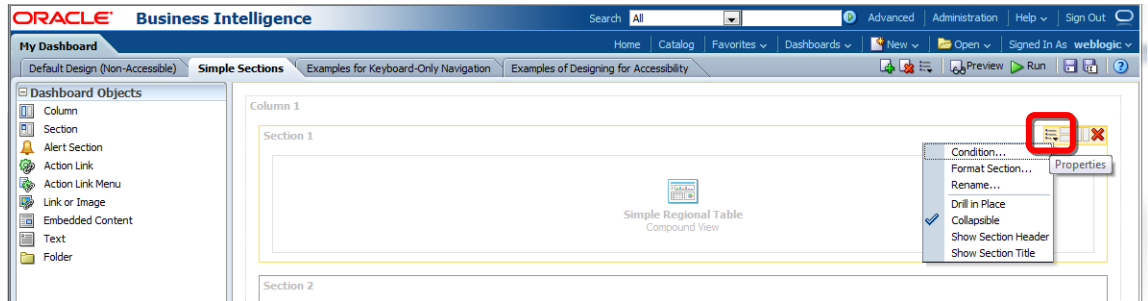
The second section, titled 'List of Offices', contains a table with the following data:

Region	Department	Office
AMERICAS	Entertainment Dept.	Blue Bell Office
	Equipment Dept.	Eiffel Office
		Perry Office
	Technology Dept.	Foster Office
	Test Programs Dept.	Casino Office
APAC	Assembled Dept.	River Office
	Local Plants Dept.	Copper Office
		Morange Office
	Manufactured Dept.	Figuerosa Office
	Translated Products Dept.	Eden Office
EMEA		Sherman Office
	Entertainment Dept.	Montgomery Office
	Manufactured Dept.	Guadalupe Office
		Mills Office
	Operations Dept.	Spring Office

Figure 18 – Default display of a multi-section dashboard

To define section names, use the following steps for every section area on your dashboards.

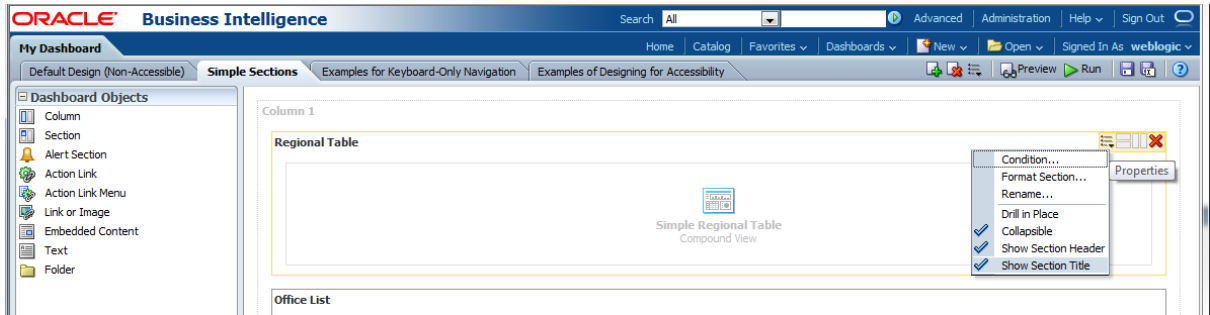
- 1) Select the “Page Options” icon on the far right of the dashboard and choose “Edit Dashboard”.
- 2) Once in the Dashboard Editor, hover over the section area and select the left-most “Properties” icon at the top of the section object.



**Figure 19 – Changing a Section’s properties**

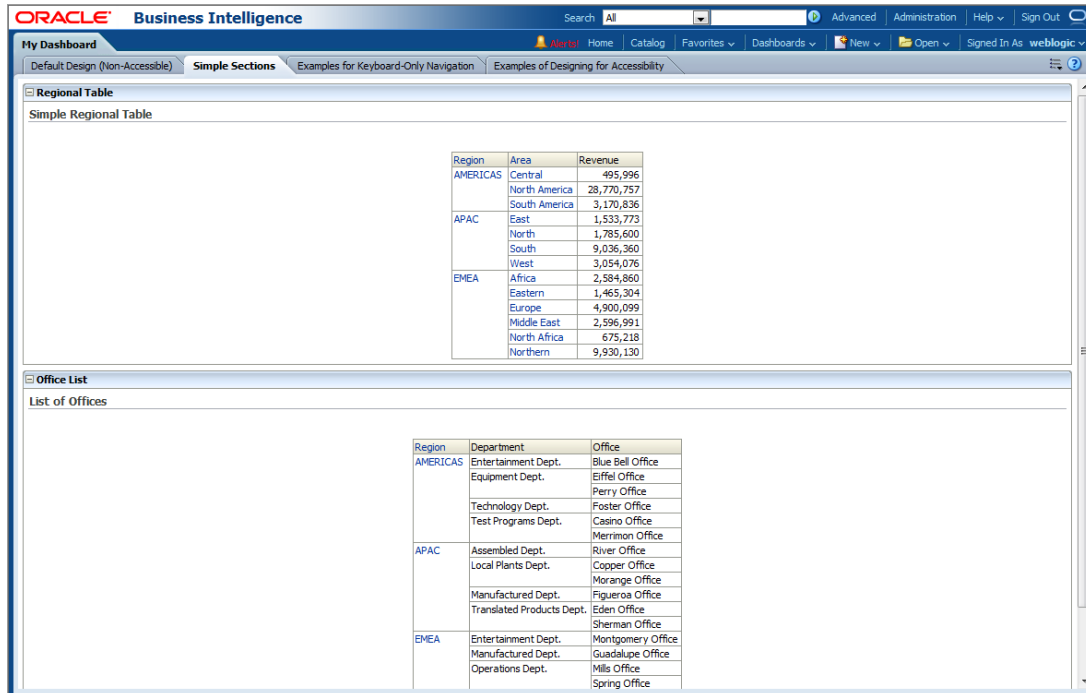
- 3) Select “Rename...” from the Properties context menu. In the dialog box that appears, type in a name that describes the contents of that section.
- 4) Select the Properties icon again and turn on the “Show Section Header” and “Show Section Title” checkboxes. Repeat this process for each section on the dashboard.

The final state within the Dashboard Editor should look something like this:



**Figure 20 – Dashboard with defined and visible Section Names**

The resulting dashboard now will look like this.



**Figure 21 – A multi-section dashboard with section names defined**

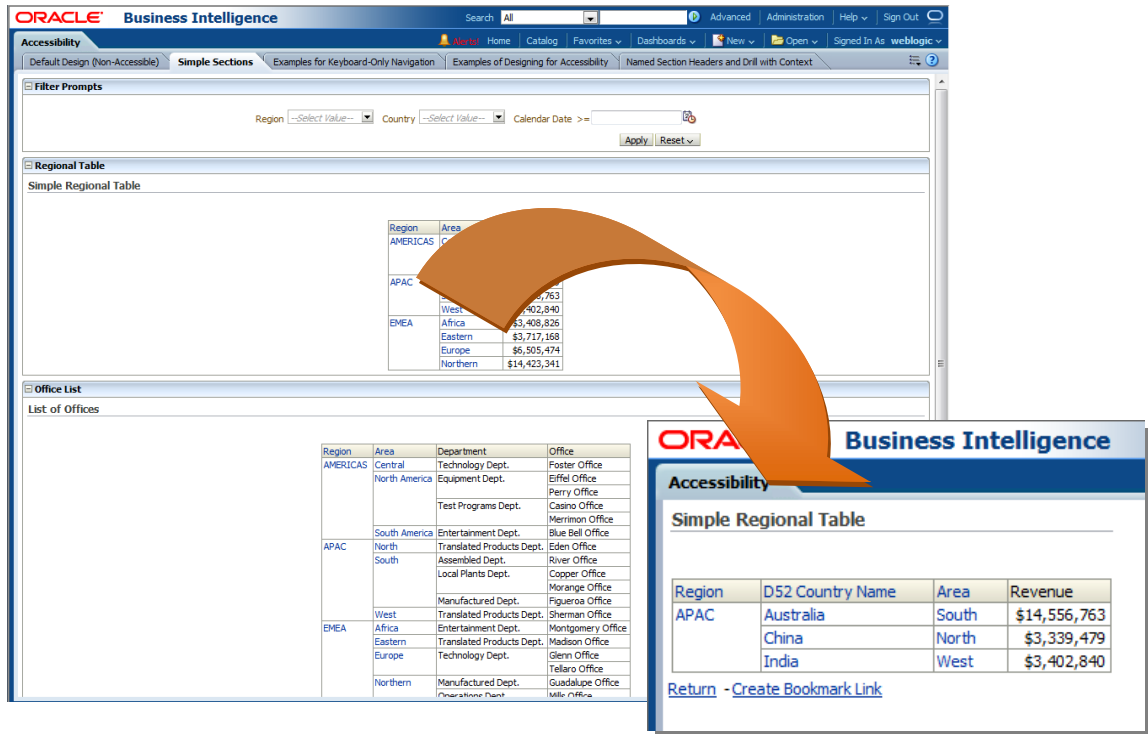
Notice that now a screen reader has enough textual information to announce the section name as well as the previously defined element names when a user navigates to them. The user will then be able to decide if they want to navigate to the elements within that section, or navigate to the next section.

In fact, these section names get surfaced to screen readers as Headers. Therefore, screen reader users will be able to rapidly navigate to specific sections in the dashboard by simply invoking the “Show Headers List” feature of their screen reader software.

### Drilling Control

Another feature available within the Dashboard Editor is the ability to control how drilling takes place on the screen. Accessibility requirements in some localities dictate that screen context remain consistent until the user invokes some action that the user would expect to change the context. As such, drilling within a table or pivot should typically keep the user on the same dashboard page on which they started. This is not the default behavior. The content design must apply the appropriate drilling control setting as part of the design of each dashboard page.

For example, the below screen shots illustrate the default behavior when drilling into a data element within a non-hierarchical column (in this case, the North area of the APAC region):



**Figure 22 – Default Drill Behavior in a Dashboard**

Notice how the default result of the drill is to display a new page that contains only a drilled version of the individual data display. All dashboard context has been removed, including any dashboard prompts that might have previously been visible to show filtering context.

To retain the dashboard context while drilling, you need to turn ON the “Drill in Place” property of each section within the dashboard.

To turn In-Place Drilling on, use the following steps for every section area on your dashboards.

- 1) Select the “Page Options” icon on the far right of the dashboard and choose “Edit Dashboard”.
- 2) Once in the Dashboard Editor, hover over each section area and select the left-most “Properties” icon at the top of the section object.
- 3) From the Properties menu, click on the “Drill in Place” item to turn it on. A checkmark should appear next to the item.

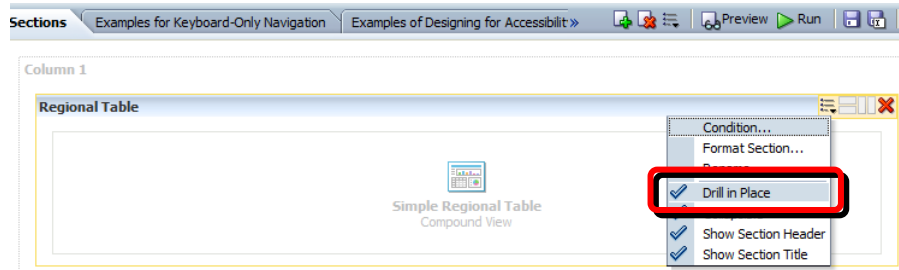


Figure 23 – Turning On the Drill-in-Place feature

- 4) Repeat these steps for every section that contains a drillable object.

The result of turning on the Drill-in-Place feature is illustrated in the following screen shots:

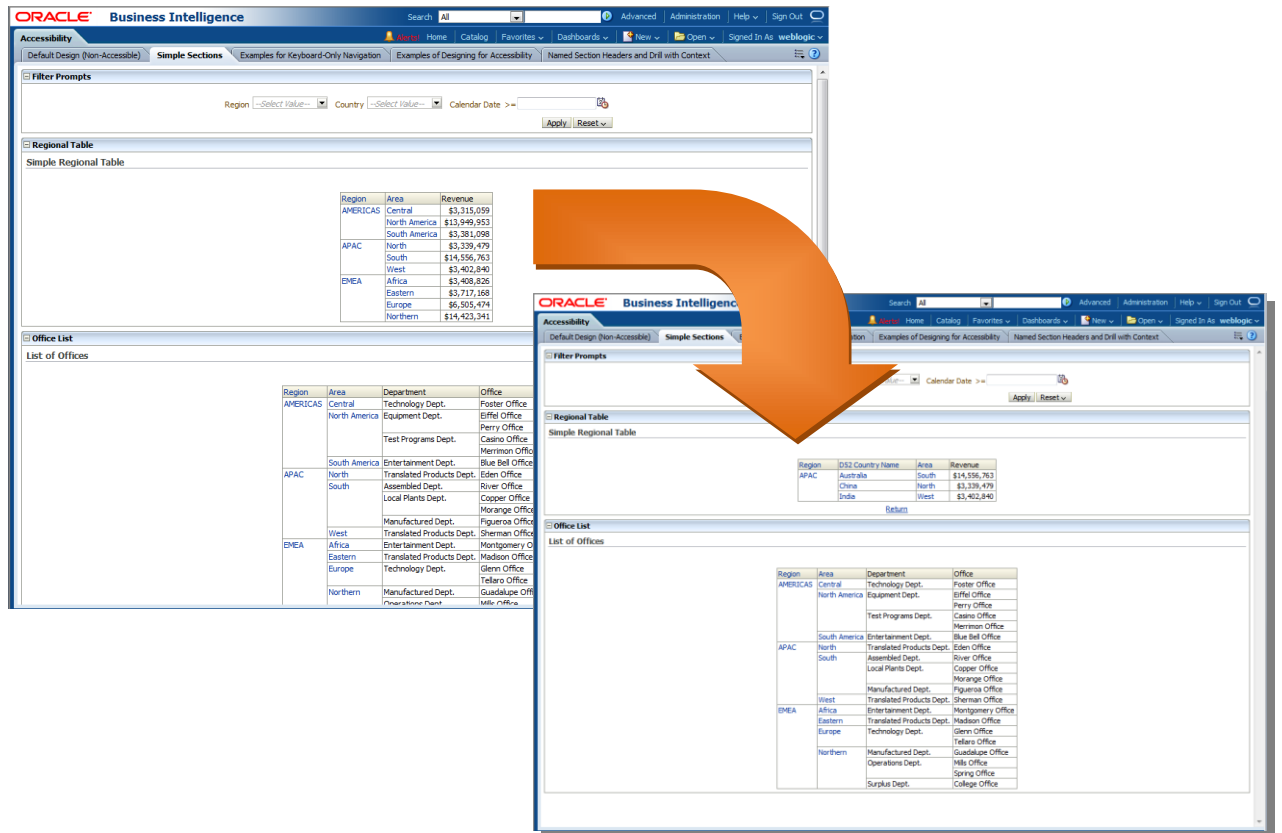


Figure 24 – The Context-Preserving Effects of In-Place Drilling

### Working with Oracle Dashboard Styles

The overall look and feel of any Oracle dashboard is controlled by a set of skins and styles that are available. You can create your own custom skins and styles to implement standard settings that support accessibility, such as default font selections, high-contrast color schemes, etc. You can set a default style for all dashboards and you can also select a style to apply to an individual dashboard.

### Creating Custom Styles

Oracle BIEE 11g ships out-of-the-box with two default styles – BLAFP (Browser Look and Feel Plus) and Fusion (FusionFX), both found in the *ORACLE\_HOME/bifoundation/web/app/res* directory. These provide a starting point for the custom style and skin that the user wishes to deploy.

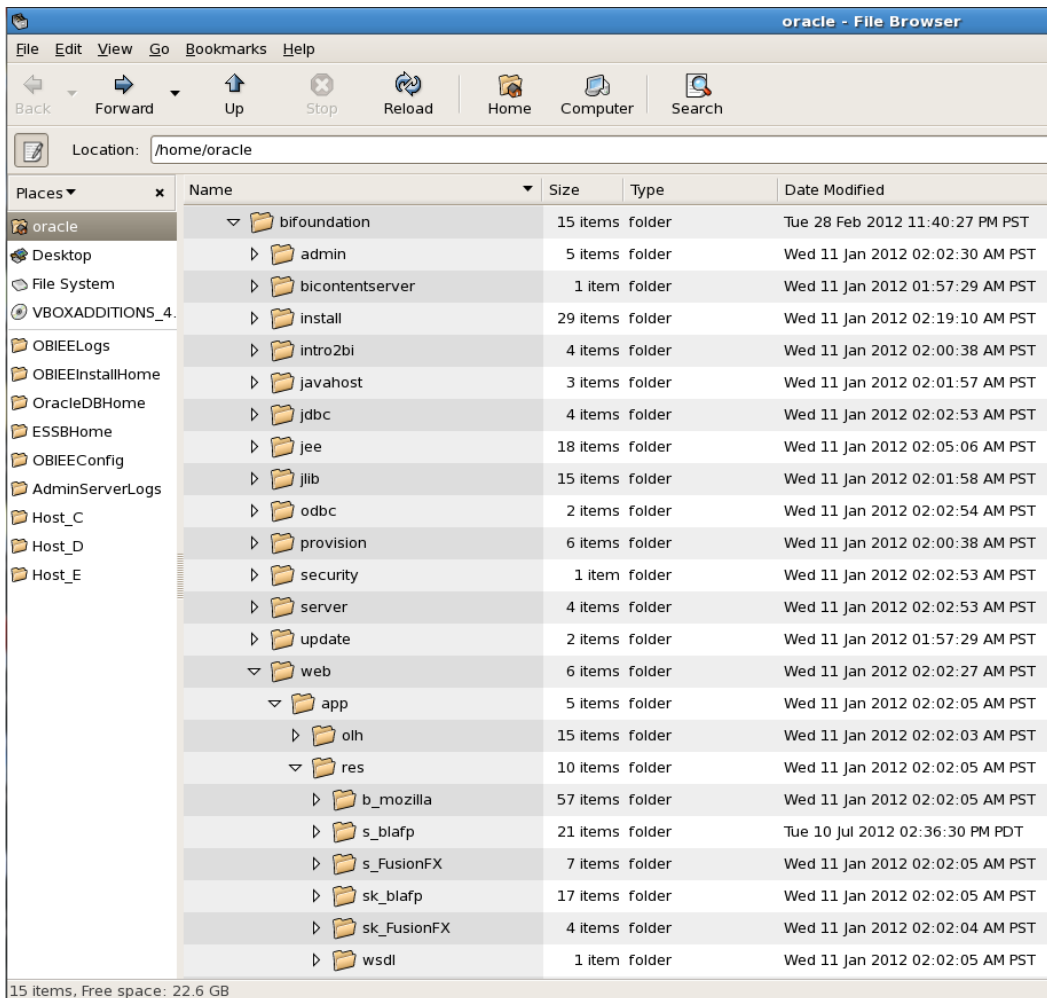
---

**NOTE:** Default styles and skins in the *ORACLE\_HOME/bifoundation/web/app/res* directory should not be modified. This directory will be overwritten with any upgrade or new installation.

---

The process to create a custom style is relatively simple. You start by copying one of the default styles into a separate folder and then you modify it to match your desired settings. By placing these copies in separate custom style folders, you preserve your custom styles across patches and upgrades.

Folders that contain Oracle Dashboard Styles are denoted with an “s\_” as the first part of the folder name. Folders that contain Skins are denoted with an “sk\_” as the first part of the folder name. The default organization of the styles shipped with OBIEE 11g is shown in the next figure.



**Figure 25 – Oracle 11g Styles and Skins folder structure**

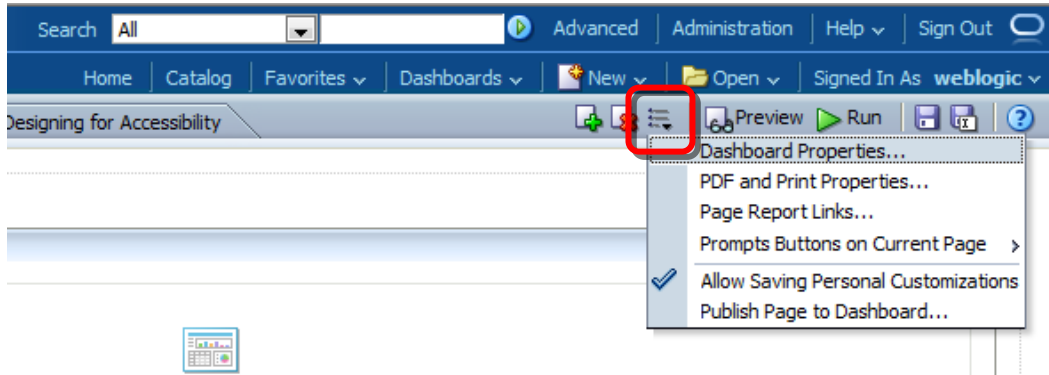
Complete details of the customization process can be found in the “Customizing Oracle Business Intelligence Enterprise Edition 11g” whitepaper, located at <http://www.oracle.com/technetwork/middleware/bi/customizing-oracle-biee-11g-176387.pdf>.

#### Applying a Style to an Individual Dashboard

To apply a custom style to one or more individual dashboards, you enter the dashboard editor for each dashboard with which you want to use the new style.

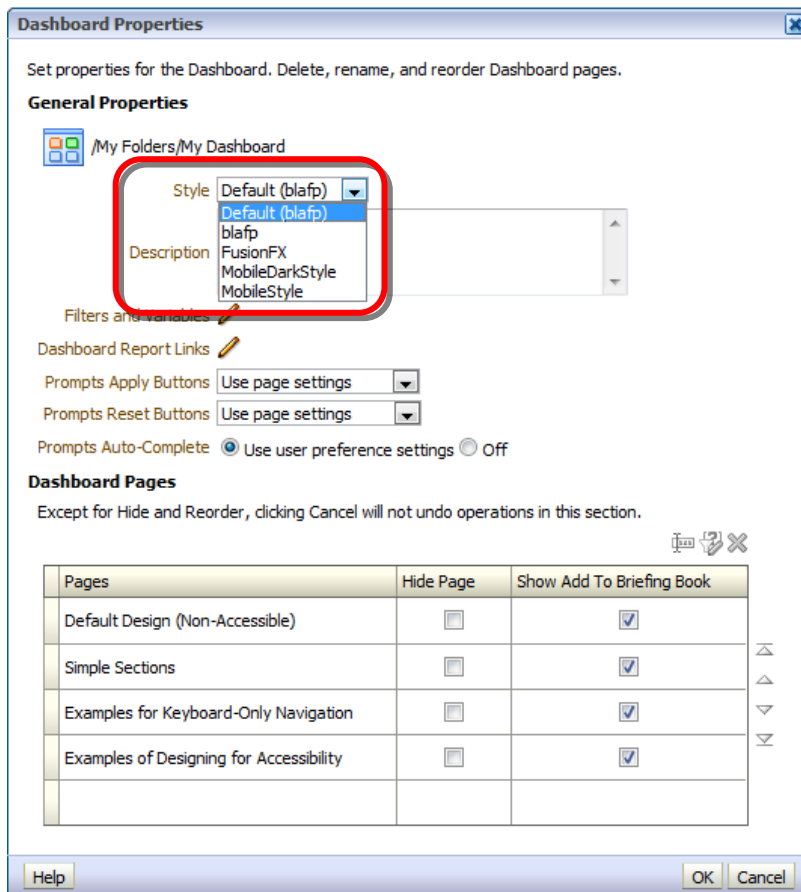
To do this, select the target dashboard, click on the Page Options icon on the right side of the screen and then select “Edit Dashboard”. Once in the main Dashboard Editor screen, select the “Tools” icon at the top of the screen just to the left of the “Preview” icon.





**Figure 26 – Accessing the dashboard properties**

The Dashboard Properties dialog box will then appear. This properties dialog will allow you to select from among the available styles, including any custom styles you have created. These will be located in a “Style” dropdown list near the top of the Dashboard Properties screen.



**Figure 27 – Applying a Custom Style to a Dashboard**

### Other Customizations

If you are planning to customize the look and feel of an OBIEE installation beyond simply creating a custom style, you need to remember that OBIEE is not a web site full of static content. Every display shown to a user is generated on-the-fly based on the user's login credentials, the user's associated security profile, the user's assigned roles, as well as the user's preferences and past usage. As such, many of the accessibility features within OBIEE are implemented in coded routines that are executed each time a new display is generated.

As a result, great care *must* be taken that these coded accessibility features are not disabled by any customizations performed on the OBIEE environment. Specifically, take care when editing existing OBIEE files outside the previously noted skins and styles. Pay special attention to any sections labeled with a "Do Not Edit" warning. You must not change these code sections in any way, nor cause them to no longer be called by any changes you make elsewhere.

### Oracle Dashboards: Dashboard Prompt Requirements

Certain design changes are needed to insure that dashboard prompts will be accessible, especially for screen reader users. Different prompt types have individual requirements for accessibility.

#### All Dashboard Prompts

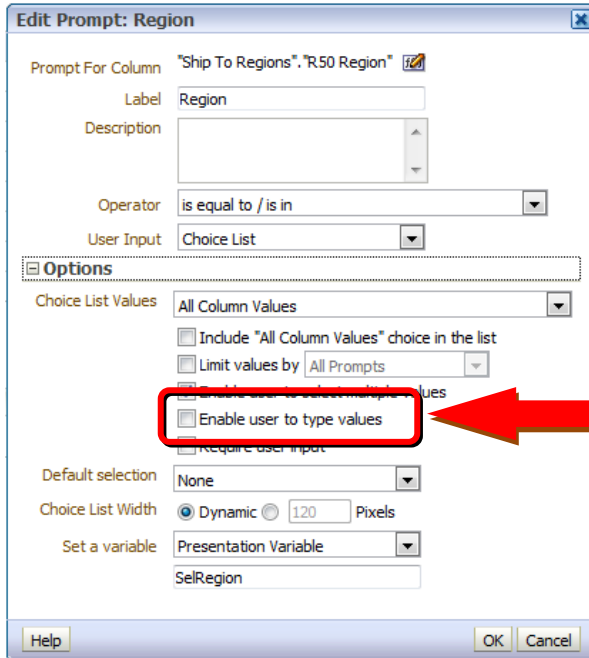
Accessibility design requires that all prompts have understandable labels associated with every element. Content designers must insure that each prompt's Label property is properly defined.

#### Choices Lists

In OBIEE 11g, a Choices List is a custom control that operates like an enhanced dropdown box. Unlike a standard dropdown box, a choices list has checkboxes for selecting each item, making multi-choice selections much easier for keyboard-only users. In addition, a choices list includes a "Search..." item at the bottom of the list that provides users with advanced search capabilities among the values in the list. This is especially helpful when a long list of values exist.

Screen readers, however, do not expect this advanced level of functionality and can get confused if the Choices List is left in its default configuration. Specifically, the ability for a user to type in a value "by hand" must be turned off, as it interferes with a screen reader's type-ahead feature.

When defining a Choices List prompt, open the Options area and turn off the "Enable user to type values" setting, as shown below.



This setting must be turned *off* to insure screen reader keyboard operations work properly.  
(Default is ON.)

**Figure 28 – Choices List: Turning off manual values entry**

### Calendar Widgets

Calendar widgets themselves are designed for accessibility. However, the operator used by the prompt will determine whether the entire filter entity is accessible. If a date-oriented prompt uses a “Between” operator, the prompt display will look like this:



**Figure 29 – Non-Accessible Date Between prompt**

Note that a label only exists for the first item in the prompt. The lack of a label for the End Date prompt selection makes this particular prompt design non-accessible.

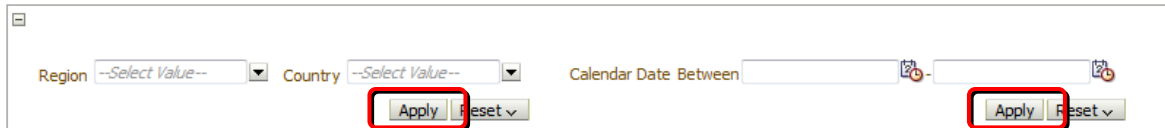
Instead of using a “Between” operator on a single date value, content designers must design their report filters and associated dashboard prompts to use a separate greater-than-or-equal-to Start Date and/or a less-than-or-equal-to End Date filter instead.



**Figure 30 – Accessible Start Date prompt**

### Use only 1 Prompt Object per Dashboard Page

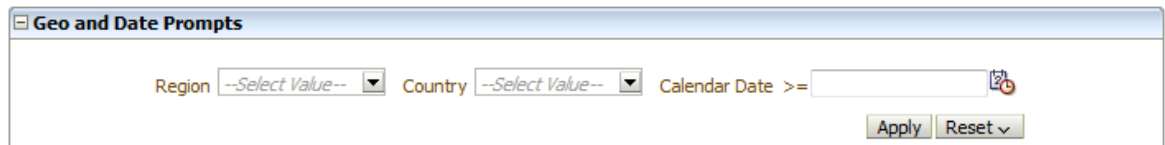
A common OBIEE design practice is to create separate individual dashboard prompt objects and then combine them as needed on a dashboard page. However, this creates problems for screen reader and voice recognition software users. The screenshot below illustrates the problem.



**Figure 31 – Multiple Apply buttons from multiple Prompt objects**

When multiple dashboard prompts appear on the same dashboard page, you end up with multiple “Apply” buttons. This presents no problem for sighted users, who can see the context of each button visually. However, screen reader users rely on special keystroke combinations to generate lists of common object types. When these lists display dashboard prompt objects, they will list multiple entries noted simply as “Apply Button”. Users of voice recognition software need to be able to speak unambiguous commands. With multiple dashboard prompts, the “Apply” buttons provide no further context and the user has no way to tell which button is which.

To eliminate this problem, you need to limit dashboard pages to use only a single dashboard prompt object. Combine all the filter selections into a single design and use this one dashboard prompt, as shown below



**Figure 32 – Single Redesigned Multi-Field Dashboard Prompt**

### Analytical Displays: Suggested Changes from Default Settings

A number of accessible designs can be implemented by leveraging certain standard but non-default display features within OBIEE dashboard objects. A few are listed here.

#### Using View Selectors to Switch Content Displays

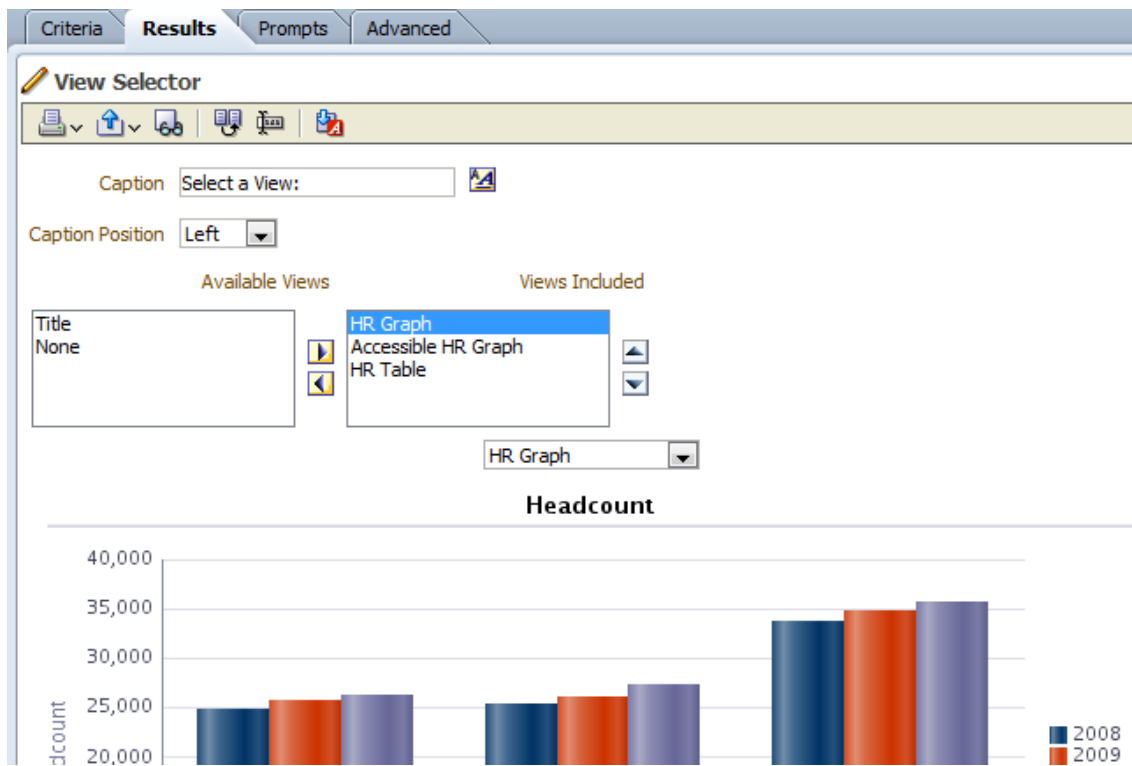
In some cases, you may want to create additional content specifically optimized for non-screen reader users with accessibility needs. The View Selector is a great way to provide the most feature-rich content for all user communities.

For example, you may want to use normal pie or bar charting most of the time, but have a cross-hatch version and a tabular version available as well to support users with color-vision issues.

You can use a View Selector and set it up as shown in the table below:

Content View	Suggested View Selector Tag
Normal Pie or Bar Chart using color	<Title> Chart
Chart with Cross-Hatching instead of color	Accessible <Title> Chart
Table	<Title> Table

Here is an example that mixes both default and accessible content equivalents in a single display using a view selector.



**Figure 33 – View Selector Definition**

Note that it is important to define a Caption property for the View Selector. Adding a suitable caption is a required design step. Otherwise, there will not be sufficient labeling to make this control accessible.

The resulting display is shown in the following figures:



Figure 34 – View Selector in Action



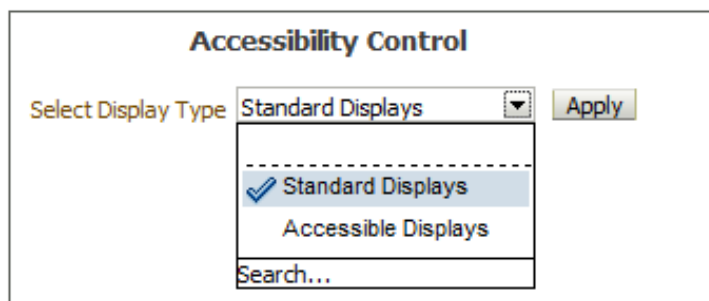
Figure 35 – After selecting accessible Chart version

In the above example, the default chart display appears first, but a user can select an equivalent chart designed to support accessibility needs by using the dropdown selector. The accessible version of the chart uses cross-hatching in addition to color to denote the different data series, and adds text elements that display the actual values each bar represents.

### Using Dashboard Prompts to Switch Content Displays

The limitation with using View Selectors to switch between non-accessible and accessible content is that they only apply to specific content displays within an individual analysis. Depending on the specifics of your application, this has the potential of increasing the amount of display design work you need to perform in every analysis you create.

An alternate approach is to create separate sets of displays for each audience, and then use a special dashboard prompt and selection queries to control the visibility of sections within an entire dashboard. In this manner, you can display completely different content in a dashboard based on accessible user selection.



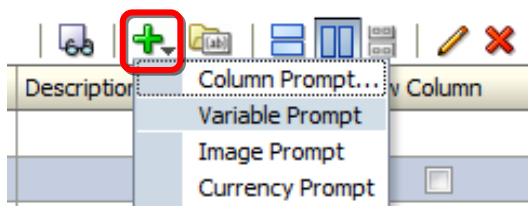
**Figure 36 – Sample Visibility Control Dropdown**

This is accomplished by using a combination of multiple Dashboard Sections, the “Conditions” properties of those sections, an Analysis request that acts as a visibility controller, and a Dashboard Prompt that assigns a value to a Presentation Variable which ties them all together. This is actually a lot easier to set up than it may initially sound.

#### 1) Create a Dashboard Prompt

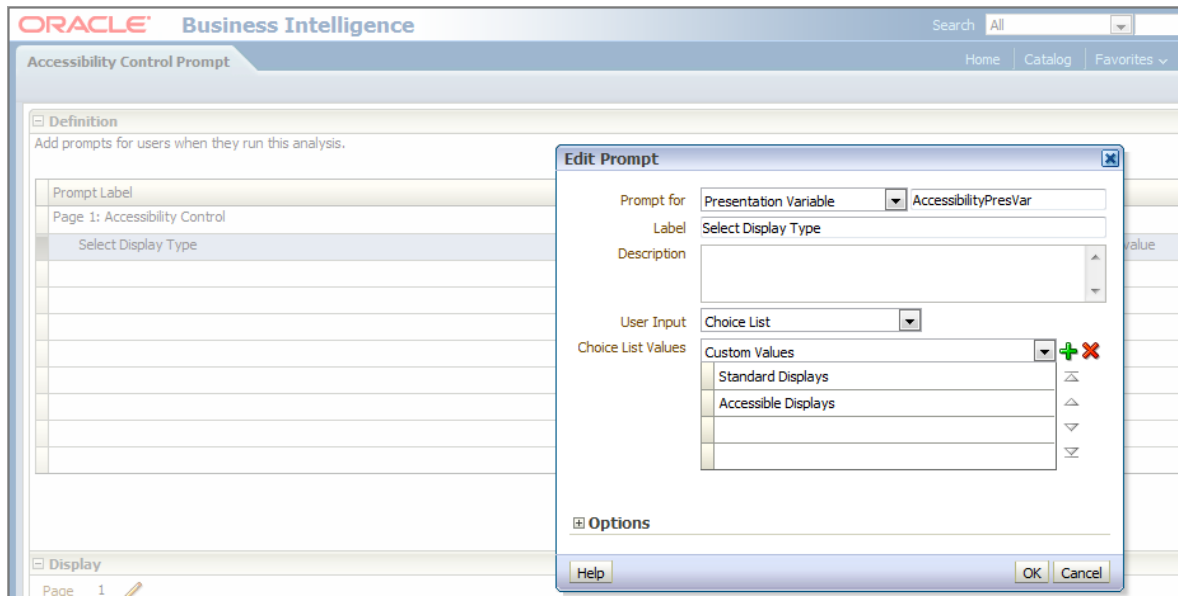
Design a Dashboard Prompt to create a Presentation Variable used to store which value the end user selects from the dropdown list. We will use a Presentation Variable named “AccessibilityPresVar”.

First, select “New”, “Dashboard Prompt” from the main OBIEE menu. Then click on the “Add” icon in the right-side toolbar to add a new Variable Prompt item:



**Figure 37 – Defining a Variable-based Prompt**

Define the prompt to use a Choices List, and click on the plus sign icon to add the Custom Values shown below:

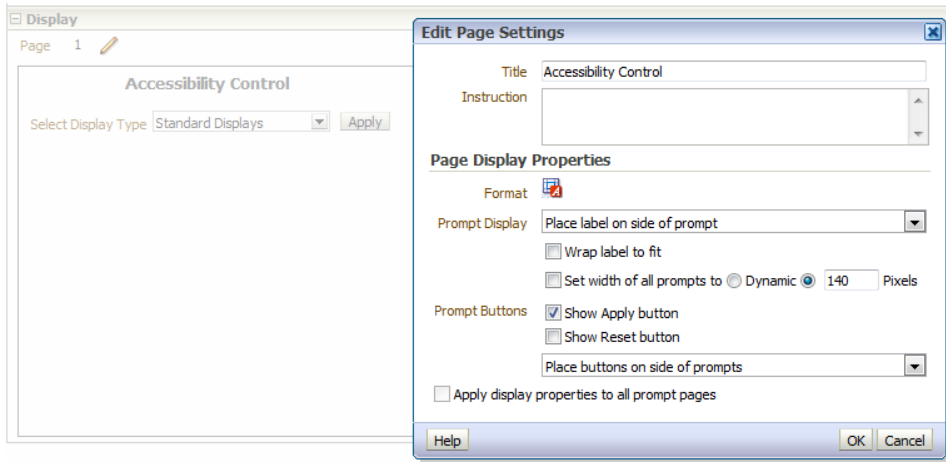


**Figure 38 – Dashboard Prompt definition using a Presentation Variable**

You also want to control the prompt’s page header and “Apply button” configuration. To do this, click on the pencil icon next to the “Page 1” header in the Display area. Edit the Page Settings as shown to set the title and prompt width, as well as the label and prompt button locations as follows:

- Title = Accessibility Control
- Set the width of all prompts to Dynamic
- Prompt Buttons: Show Apply Button = On
- Prompt Buttons: Show Reset Button = Off
- Prompt Buttons: Place buttons on side of prompts



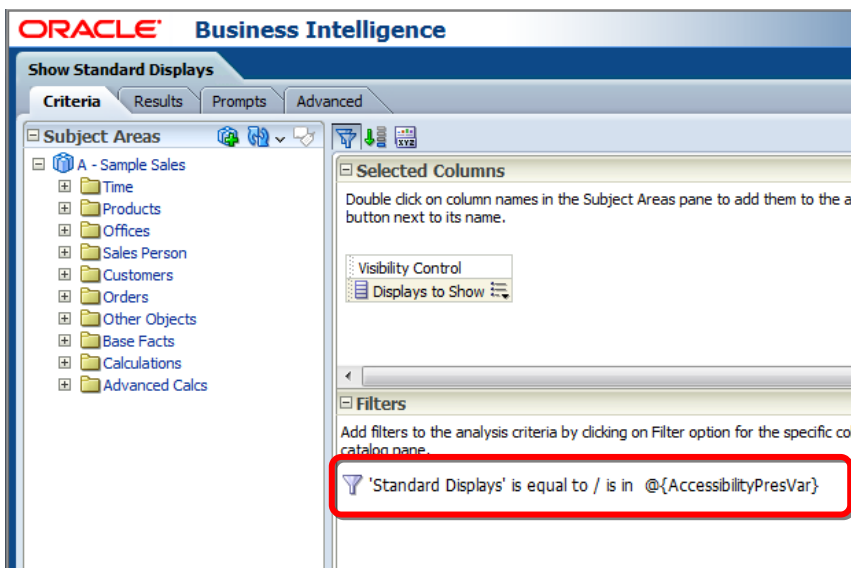


**Figure 39 – Defining the Prompt Page Settings**

A good “best practices” recommendation is to place all your visibility control objects into a separate folder. In this example, we saved this dashboard prompt as “Accessibility Control Prompt” into a folder named “Visibility Controls”.

**2) Create a “Visibility Controller” Analysis Request**

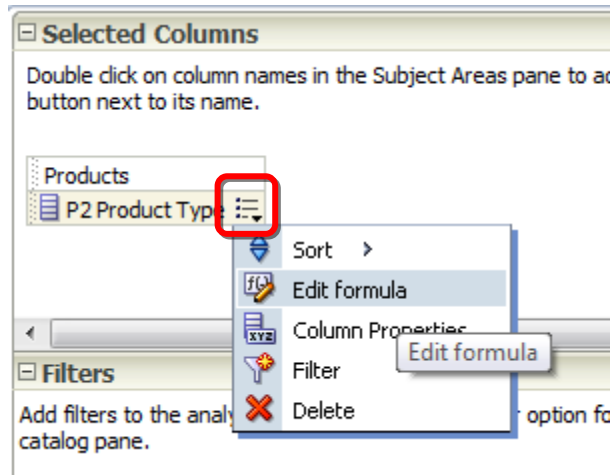
Next, you create an Oracle Analysis request that uses the value of the Presentation Variable as a filter to control whether the query request returns rows. The presence or absence of rows in this request will control which set of displays are made visible. Below is an example controller request that will return one row if the user selects a value of “Standard Displays” from the dropdown, and will return no rows if they select “Accessible Displays”.



**Figure 40 – Analysis Request as Visibility Controller**

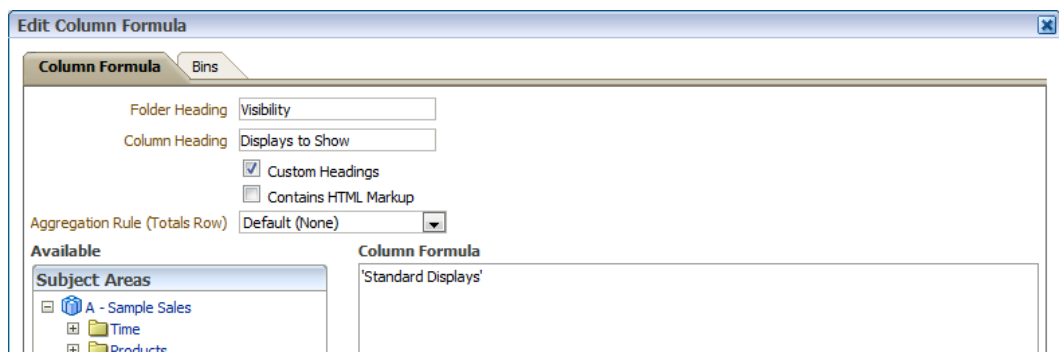
To define this Visibility Control Analysis Request, use the following steps:

- a) From the main OBIEE menu, select New | Analysis.
- b) Select the same Subject Area you used to create the “Accessibility Control Prompt” Dashboard Prompt in Step 1.
- c) Select any data column in the Subject Area and add it to the request. Which column you use doesn’t matter, since we will be modifying the column formula anyway.
- d) Click on the column properties menu icon and select “Edit Formula”.



**Figure 41 – Modifying an Analysis column’s formula**

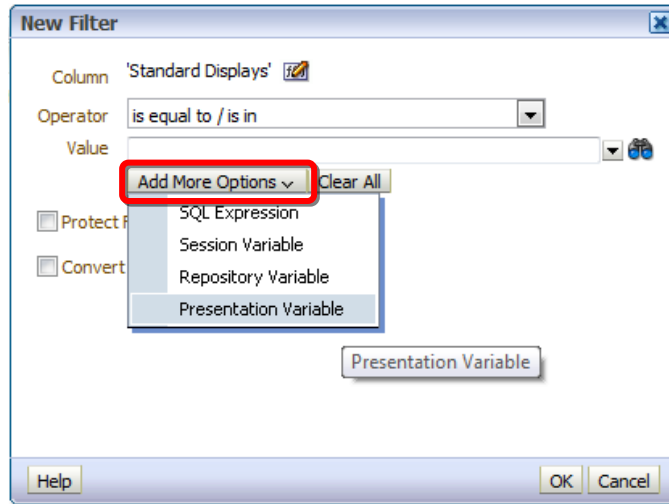
- e) Turn on the “Custom Headings” checkbox and change the column headers as well as the formula as shown. Note the formula’s use of single quotes to enclose the string used in the dashboard prompt for selection of the default non-accessible displays.



**Figure 42 – Defining a custom formula to control visibility**

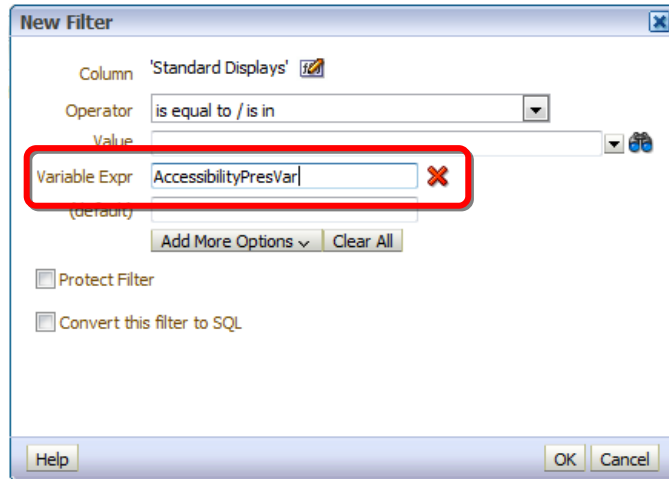
The result of these modifications is to create an analysis that returns a single row with a single value column that contains the literal string “Standard Displays”. We will next define a filter that will conditionally display or hide this data row.

- f) Once back at the Criteria tab, add a filter to the Request on the item named “Standard Displays”.
- g) Click on the “Add More Options” button and select “Presentation Variable to tie this filter to the current value of the Presentation Variable controlled by the dashboard prompt created in Step 1.



**Figure 43 – Add a Filter based on a Presentation Variable**

- h) Enter the name of the Presentation Variable defined in the Visibility Control Prompt, AccessibilityPresVar, in the “Variable Expr” field.



**Figure 44 – Defining which Presentation Variable to use**

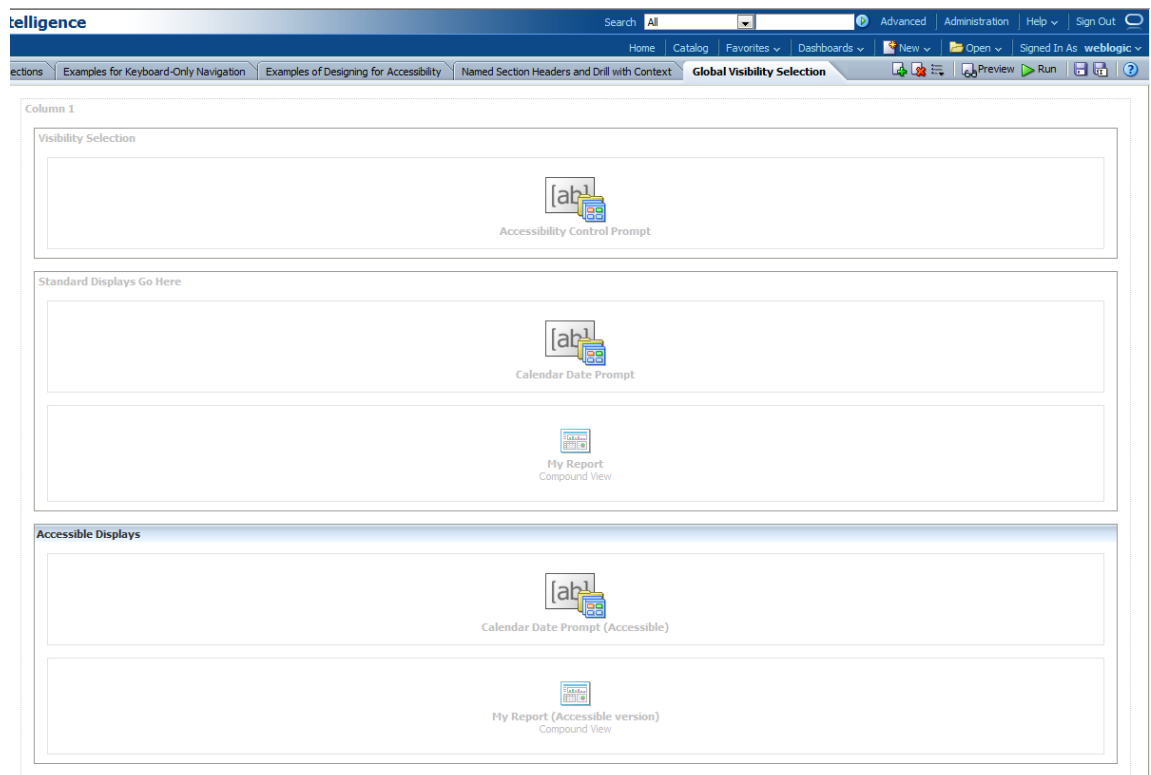
The final state of the analysis should look like what is shown in Figure 40.

- i) Save the analysis in the “Visibility Controls” folder as “Show Standard Displays”.

### 3) Create content and add it to the Dashboard

Create different reports, charts, tables, etc. for each of your display selection options. You then place all of them vertically in your dashboard design, each in their own section. At the top of this display “stack”, you place your dashboard prompt display selector. Underneath this Controller section, add a section for the standard, non-accessible content designs and a third section to hold the accessible content displays. Make sure you set the section properties for the section containing the accessible content to rename and display the section header.

Your layout should look something like this:

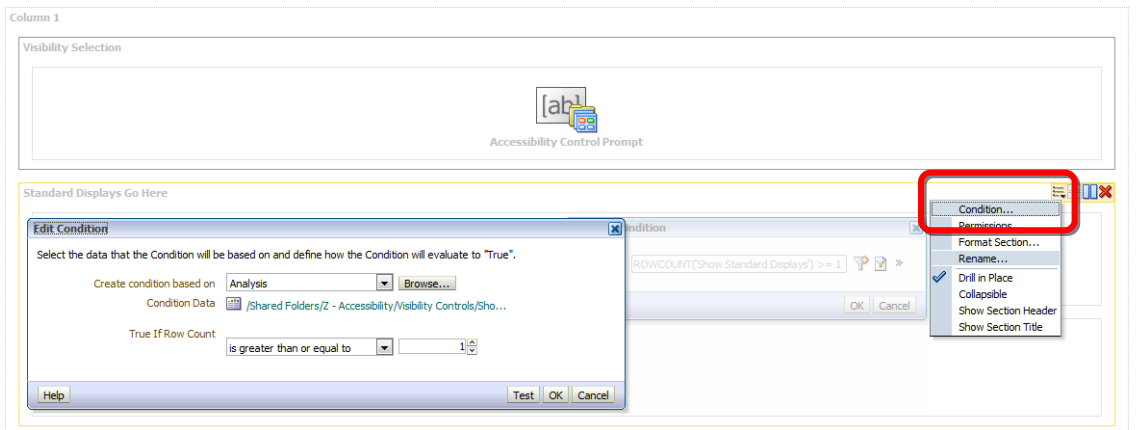


**Figure 45 – Dashboard Section Layout Organization**

### 4) Turn each display section into a Conditional Section

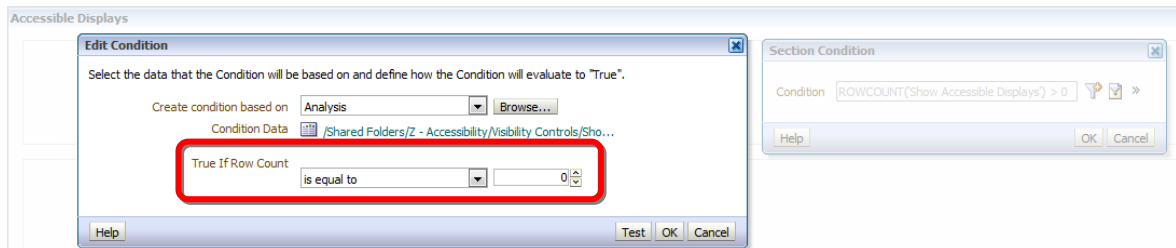
In most cases, sections simply get displayed whenever a dashboard is viewed. However, one of the properties of a section is the ability to control its visibility based on whether or not a controller query returns any rows. These are known as Conditional Sections and are defined by the “Condition...” property of the section. This property allows you to tie the visibility of the section to whether the controller request you created earlier returns rows, which in turn is based on the value selected in the dropdown for the Presentation Variable.

For the section which you want to be visible when the user selects “Standard Displays”, you click on the Section Properties button and then select “Condition...”. In the subsequent dialog box, click on the “New Condition” icon and define the condition to use the controller Analysis request named “Show Standard Displays” and set the “True If Row Count” property to “is greater than or equal to 1”, as shown below.



**Figure 46 – Making the section's display conditional**

For the display section that you want displayed when the user selects “Accessible Displays”, repeat the above process for making a section conditional, but set the “True If Row Count” property to “is equal to 0”.



**Figure 47 – Setting the Accessible Displays section's Display Condition**

The end result will be a dashboard that switches all displays at once when the user selects a different display type in the Visibility Control Prompt.

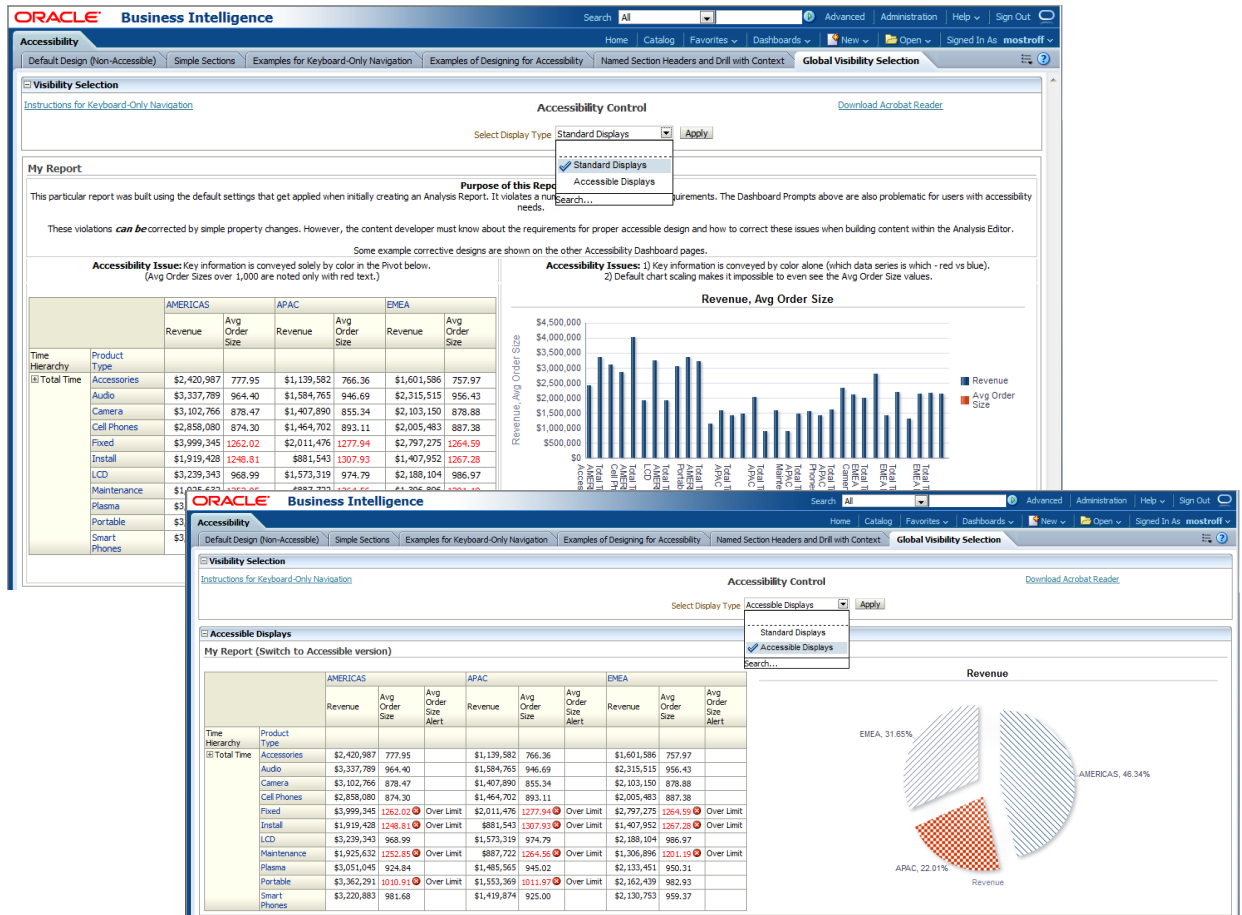


Figure 48 – Switching between Standard and Accessible Displays

### Using a System Variable to Switch Content Displays

The above technique of using a dashboard prompt to allow users to select between a set of conditional displays works well for sighted accessibility users. However, for screen reader users running in Accessibility Mode, you may want something more automatic.

The OBIEE system provides a number of automatic session-based Presentation Variables. Starting in the v11.1.1.6.7 release, a new variable has been added named **session.accessibility**. This is a logical variable that has a value of ‘true’ when the current session is running in Accessibility Mode. The value will be ‘false’ when running in regular mode.

You can leverage this automatic variable to modify the previous display switching technique to now switch conditional content automatically based on the current mode, eliminating the dashboard prompt.

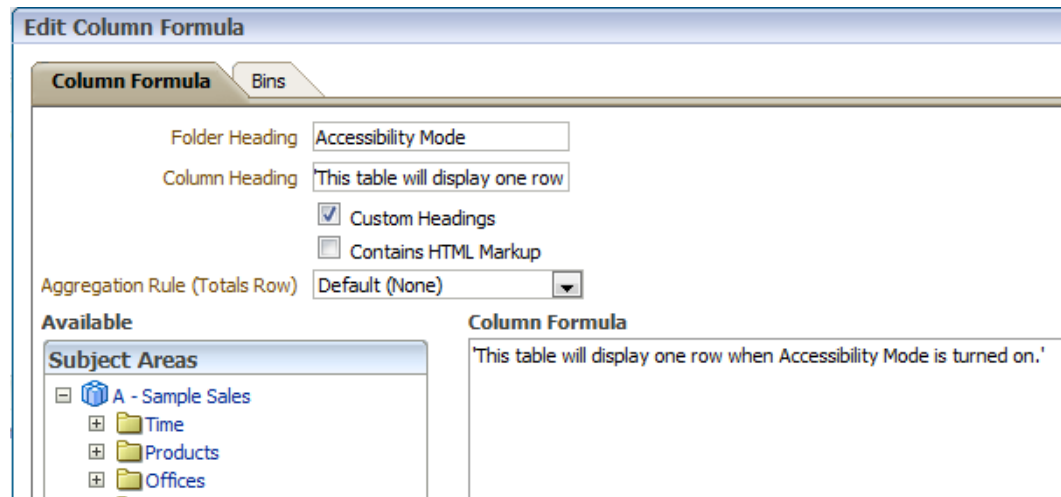
To implement this Accessibility Mode auto-switching behavior, follow these steps.

**1) Create an “Accessibility Mode Status” Analysis Request**

This analysis request will be based on the new session.accessibility variable and act as the conditional control for displays. You want to design this so that it contains 1 row when session.accessibility is true, and no rows when the variable is false. You may also want the Accessibility Mode status to be displayed as a message at the top of your dashboard, so the design should also include a static text object that displays the value of the variable. (You need a static text object because the table will display a “No Rows Returned” error message in regular mode, which might confuse your users with an “error message” about something that is not really an error.)

To define the Accessibility Mode Status request, use the following steps:

- a) From the main OBIEE menu, select New | Analysis.
- b) Select any Subject Area.
- c) Select any data column in the Subject Area and add it to the request. Which column you use doesn't matter, since we will be modifying the column formula anyway.
- d) Click on the column properties menu icon and select “Edit Formula”.
- e) Change the Folder Heading, Column Heading and Column Formula as shown below.



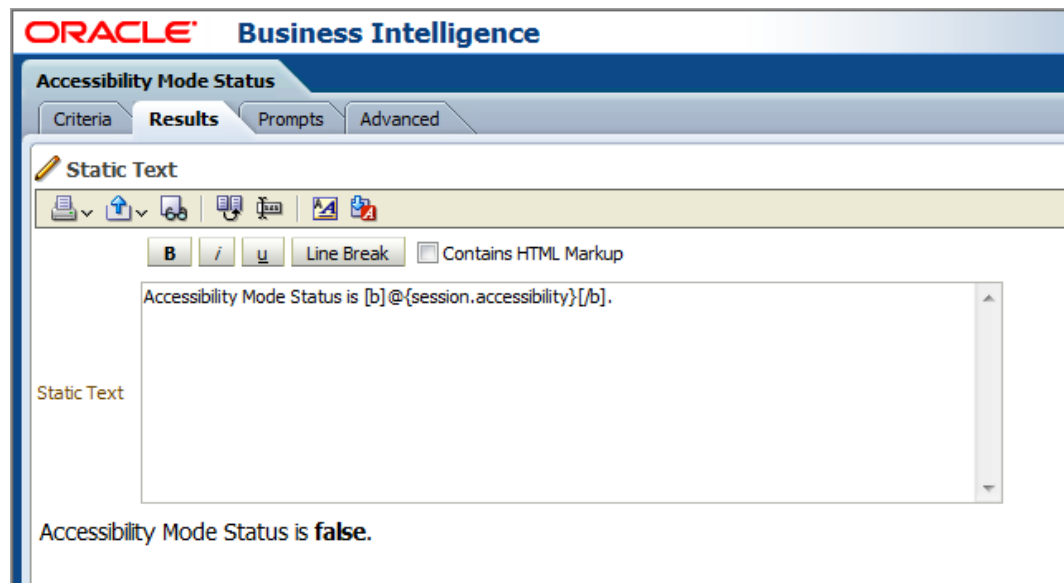
**Figure 49 – Accessibility Mode Status: Column 1 Formula**

Parameter	Value to Use
Folder Heading	Accessibility Mode
Column Heading	‘This table will display one row when Accessibility Mode is turned on’
Column Formula	‘This table will display one row when Accessibility Mode is turned on’

- f) Repeat the above procedure to add a second column. This time, define the column formula as follows:

Parameter	Value to Use
Folder Heading	Accessibility Mode
Column Heading	'true'
Column Formula	'true'

- g) Select the Results tab, and then add a new Static Text object to the display. This will be used to create a message display for the current accessibility mode status.
- h) Click on the pencil icon on the Static Text object and define it as shown below. (You may want to use the format icon to increase the size of the display font to make the message easier to read. The below example has the font set to 14 pt.)



**Figure 50 – Accessibility Status Mode: Static Text Object as Status Message**

The proper Static Text to use is:

Accessibility Mode Status is [b]@{session.accessibility}[/b].

The @{xxx} curly braces syntax is the standard way to reference a system variable.



At this point, your Accessibility Mode Status display should look something like this:

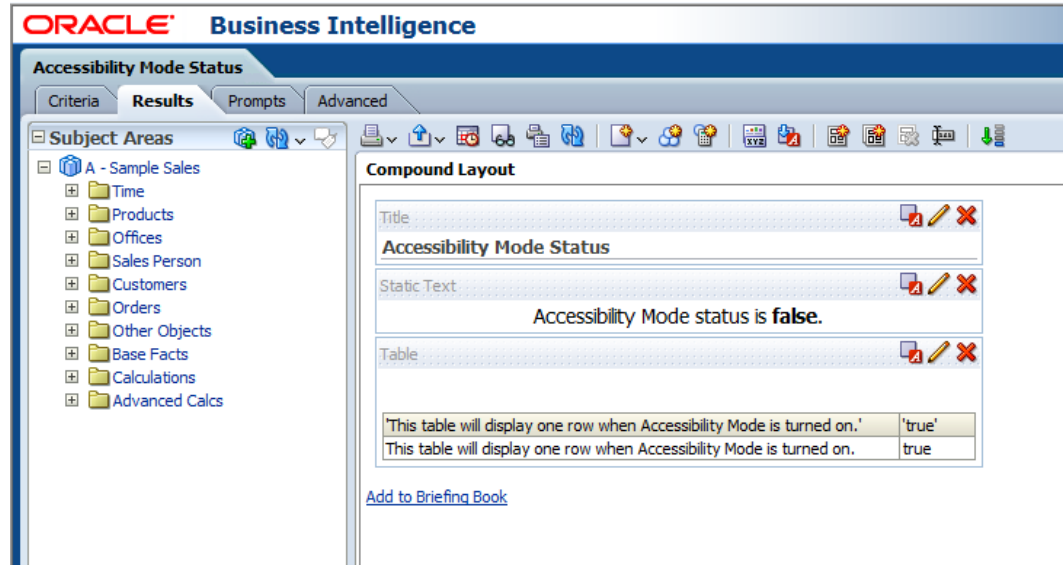


Figure 51 – Accessibility Mode Status: Unfiltered display

- i) Next we need to add a filter on the ‘true’ column, based on the system presentation variable.
- j) Select the Criteria tab and add a filter to the ‘true’ column. In the New Filter dialog box, click on the “Add More Options” button and select “Presentation Variable”.

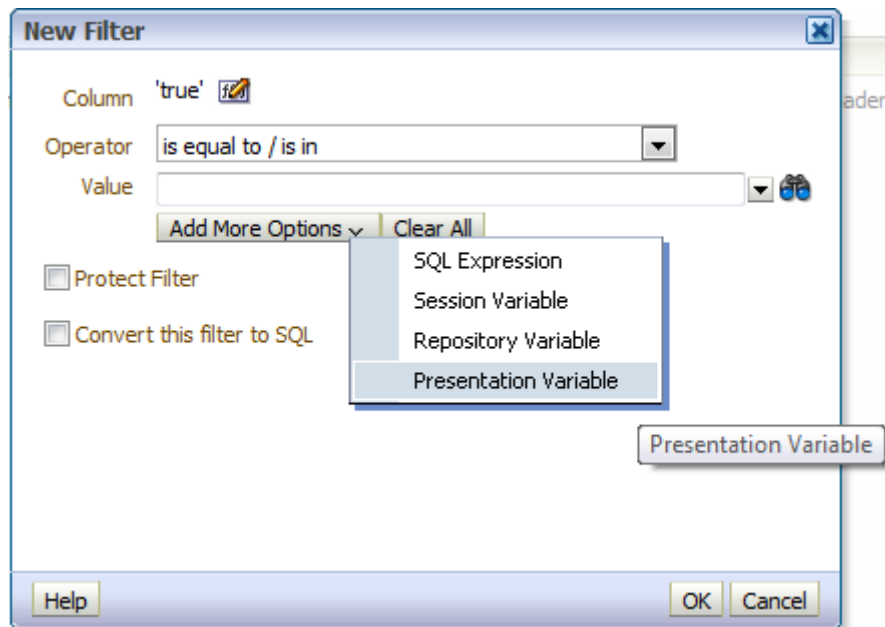


Figure 52 – Accessibility Mode Status: Creating a Presentation Variable Filter

- k) Define the filter as shown below, using only the variable’s name without the curly braces:

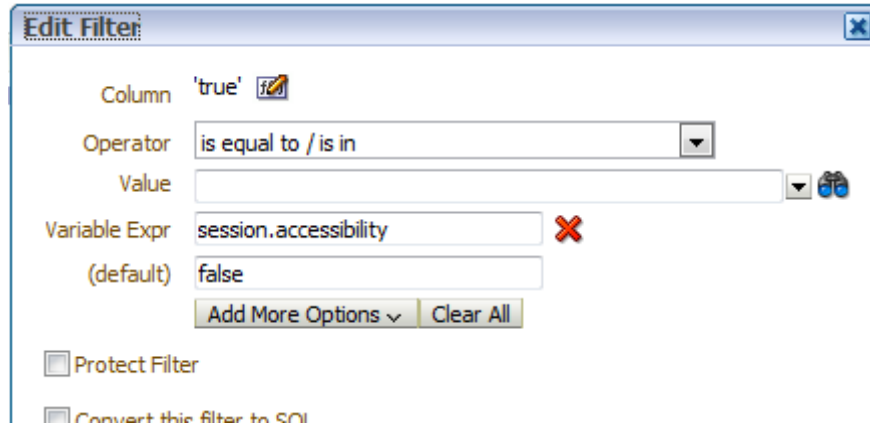


Figure 53 – Accessibility Mode Status: Defining the Filter Condition

- l) The resulting analysis should display as shown below:

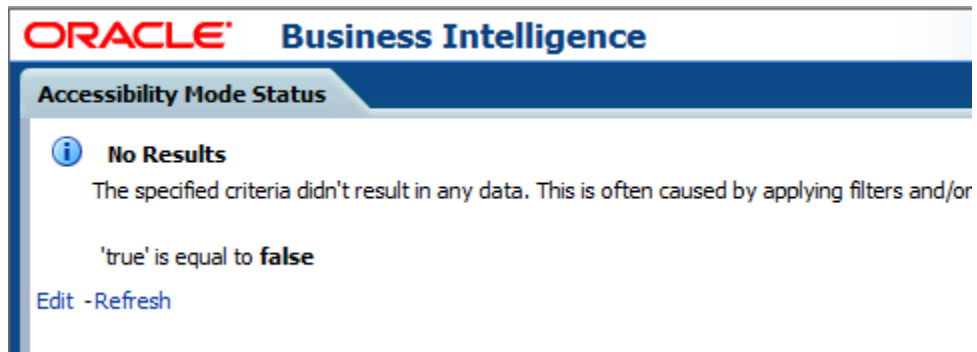


Figure 54 - Accessibility Mode Status display in Regular Mode

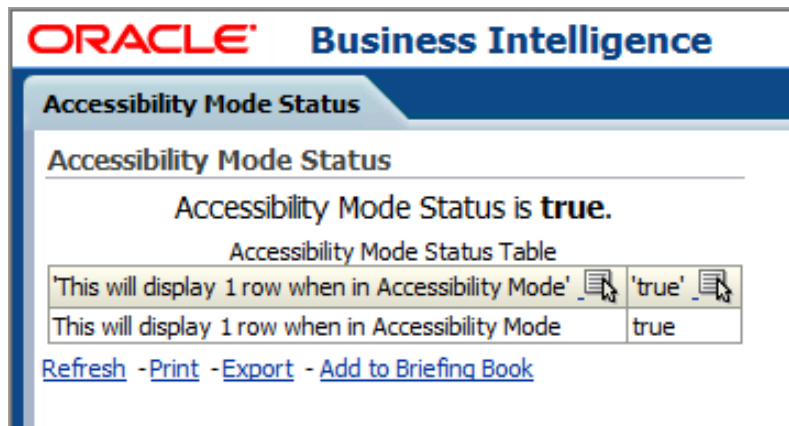


Figure 55 - Accessibility Mode Status display in Accessibility Mode

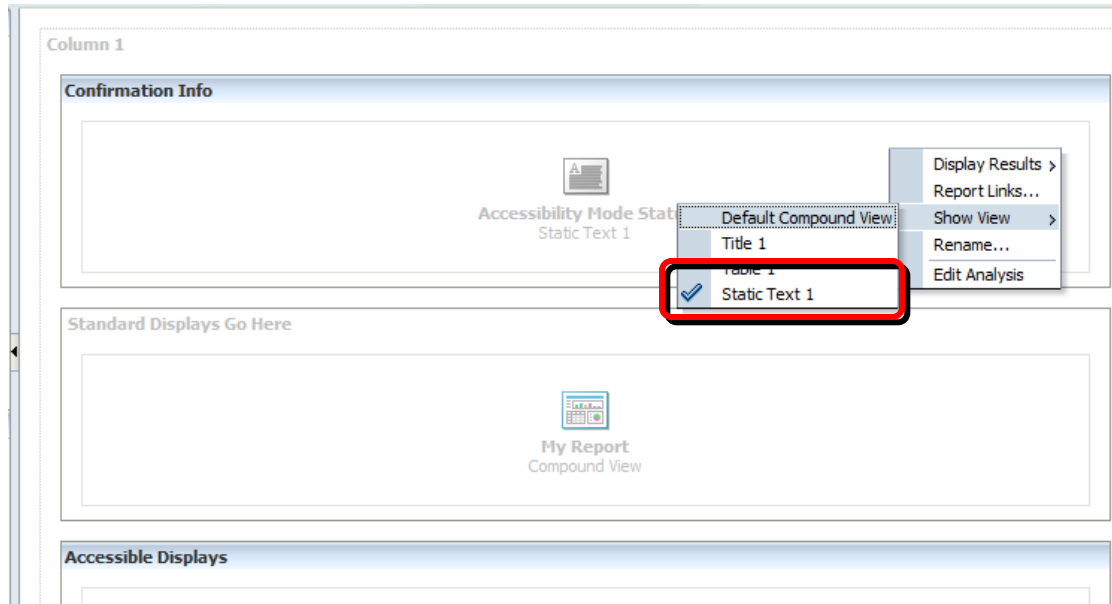
**2) Create content and add it to the dashboard**

Create at least 3 sections to hold your dashboard content. You will need 1 section to hold content to display to all users, a second section to hold content displayed only to regular mode users and a third section to hold content to be displayed only to Accessibility Mode users.

Make sure you define section names and make the section headers visible for the “All Users content” and “Accessibility Mode content” sections. Add your content as appropriate.

**3) If desired, add a display of the Accessibility Status to the top of the dashboard**

In the “All Users content” section, add the Accessibility Mode Status analysis. To avoid displaying a false “error message” of No Rows when in regular mode, change the View to be just the Static Text object view instead of the default Compound View, as shown here:



**Figure 56 – Set the view to display only the Static Text object as a status message**

**4) Turn each display section into a Conditional Section**

Make sure both the “Regular Mode content” section and the “Accessibility Mode content” section are set to be conditional displays. Both should be based on the Accessibility Mode Status analysis created in step 1 above.

Section	Analysis to Use	Condition
Regular Mode content	Accessibility Mode Status	Row Count is Equal to 0
Accessibility Mode content	Accessibility Mode Status	Row Count is Greater than 0

The resulting dashboard display should look something like this:

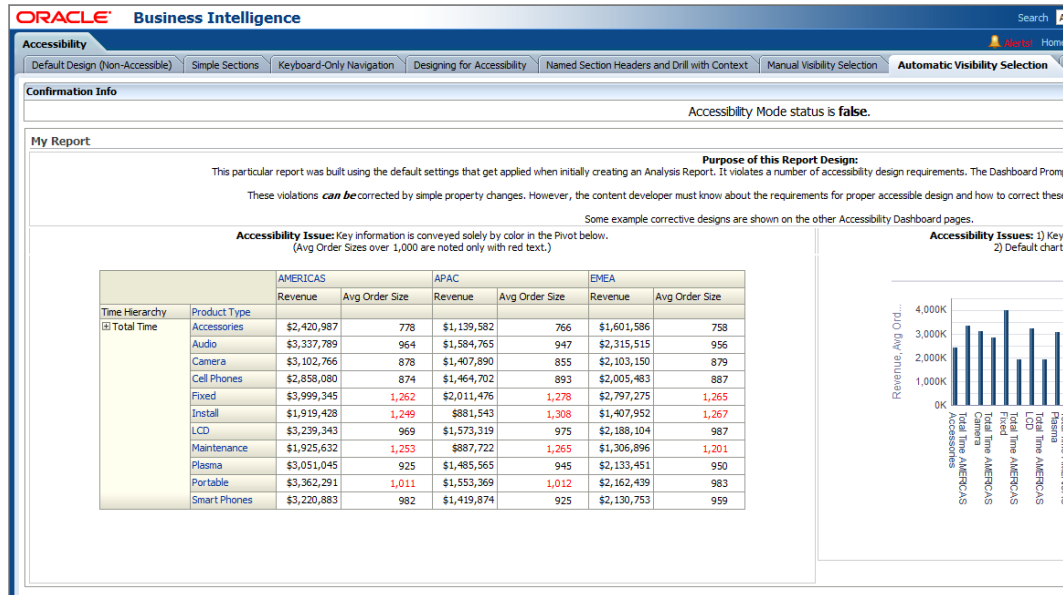


Figure 57 - Auto-switching Dashboard in Regular Mode

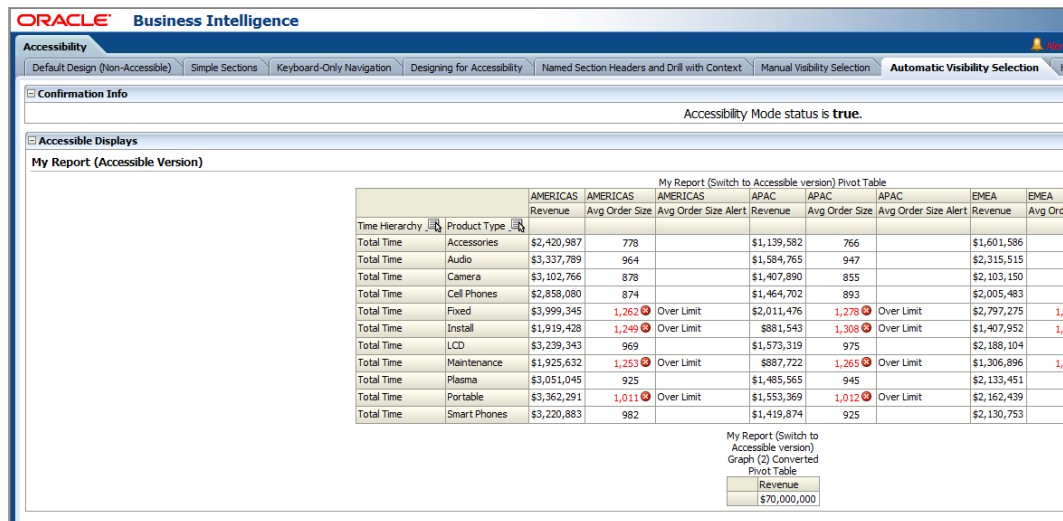


Figure 58 - Auto-switching Dashboard in Accessibility Mode

Notice that the report used in Accessibility Mode is not just an accessible rendering of the same report as displayed in Regular Mode. This report adds a new “Avg Order Size Alert” message column to the display to provide an alternate method besides color of noting a data value alert.

### Special considerations for Exalytics deployments

If you use the keyboard to navigate among the values within a dropdown that has no APPLY button, the dropdown will automatically select the next value to which you navigate. Many users with accessibility needs prefer that dropdown selection controls not execute any actions until some kind of action button has been selected. This makes navigation of dropdown lists easier to use with a keyboard.

### OBIEE running on the Exalytics In-Memory BI Machine

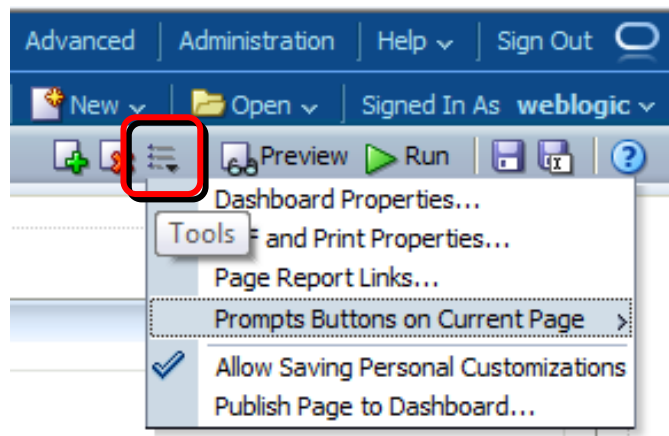
By default, OBIEE on non-Exalytics hardware always defines dashboard prompts so that they have an APPLY button. However, OBIEE running on the Exalytics In-Memory BI Machine engineered system will set up dashboard prompts to immediately apply settings changes with no APPLY button used.

For accessibility purposes, you must turn off this behavior for all dashboard prompts when running on an Exalytics machine. Two ways exist to accomplish this requirement.

#### **Method A – On a Dashboard by Dashboard Basis**

To do this, follow these steps on every dashboard that will be accessed by users with accessibility requirements:

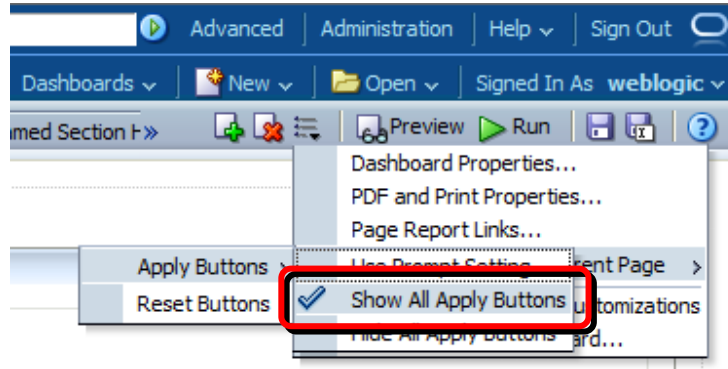
- 1) Enter the dashboard editor on the target dashboard.
- 2) Select the overall Dashboard Tools menu from right side of the Dashboard Editor toolbar:



**Figure 59 – The Dashboard Tools menu**

- 3) Select the “Prompts Buttons on Current Page” menu.

- 4) Select the “Apply Buttons” sub-menu and turn ON the “Show All Apply Buttons” setting.



**Figure 60 – Turning on all APPLY Buttons**

#### **Method B – Within the Definition of the Dashboard Prompt**

Each dashboard prompt has page settings that can be used to control whether an Apply and/or Rest button will appear on the prompt. Figure 39 above shows an example of the dashboard prompt page settings dialog. This method has the advantage of being a single location to control the use of the Apply button for a prompt, regardless of the number of dashboards on which it is used.

#### **Use Static Text objects to provide custom HTML Markup**

Screen readers usually key off of standard HTML markup to provide information for how to navigate a screen. One of the most commonly used markups is the use of Header tags. The default Title object in an Oracle Dashboard will include a TITLE tag when Accessibility mode is turned on. However, you cannot add other tags to the design of a Title object.

You can use the Static Text object to replace the Title object and supply any needed HTML tag at the same time. Simply turn on the “Contains HTML Markup” checkbox and enter the appropriate HTML code. You can even access the styles contained in your OBIEE style sheets to insure consistency with the rest of your dashboards.

For example, you can use this technique to create an HR Report’s header that tags the Title as H3 text instead of as Title text, with a horizontal rule line underneath. Simply replace the Report Title object with a Static Text object that has its text defined as:

```
<H3>HR Report</H3><HR WIDTH=650 ALIGN=LEFT>
```

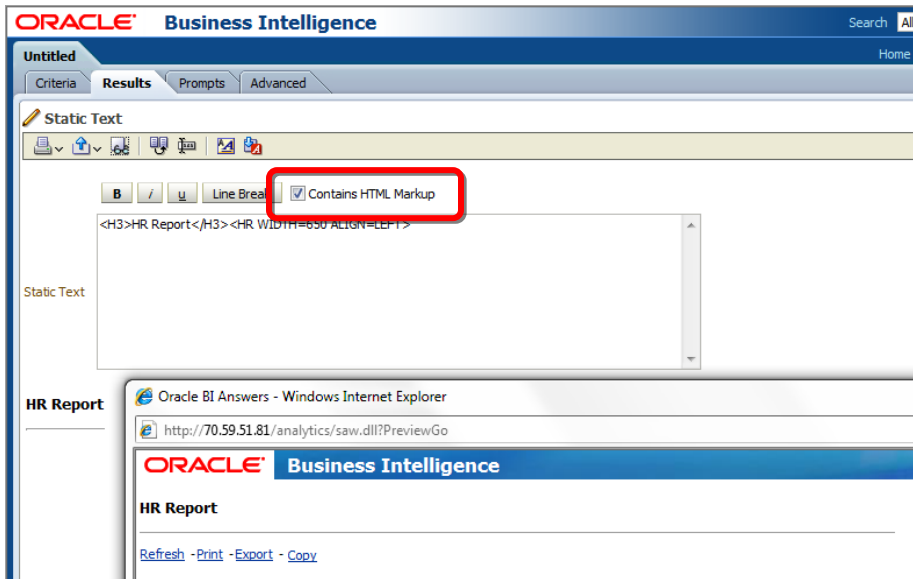


Figure 61 – Using HTML Mark-Up in a Static Text object

### Define Descriptions for all Reports

Oracle Dashboards will generate explanatory text for objects based on their description fields. Make sure each report design you create includes a short description of its functionality. You define this when you save the report object, as shown below.

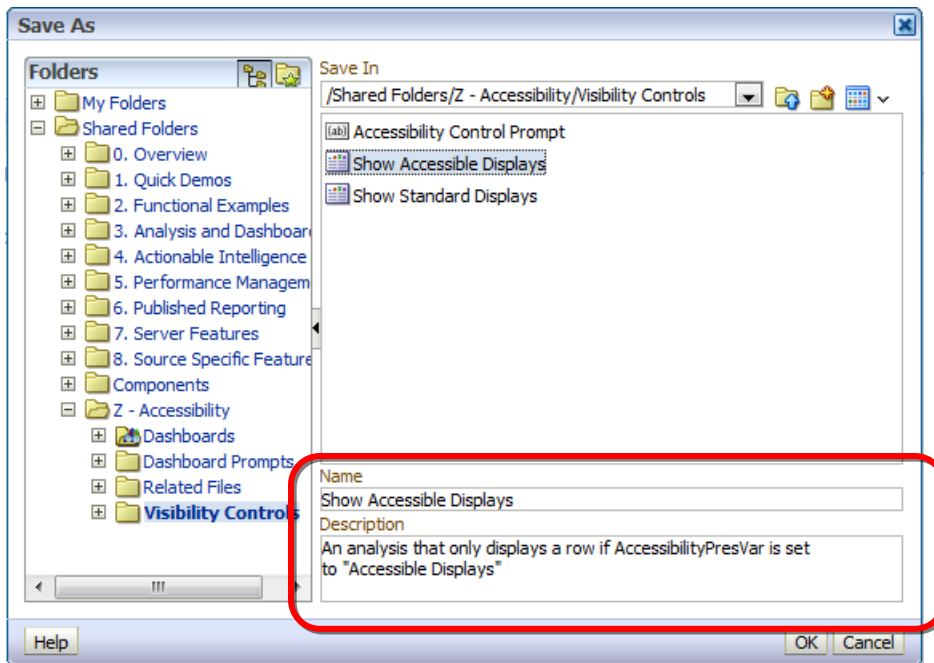
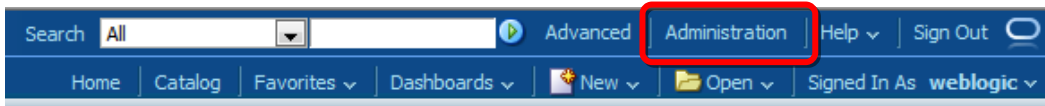


Figure 62 – Defining a Report's Description

## BI Publisher: Turning on Accessibility Mode for Output

In addition to a user turning on their Accessibility Mode setting, an OBIEE administrator must set up the BI Publisher Runtime Configuration to generate report output that is accessible to users with disabilities.

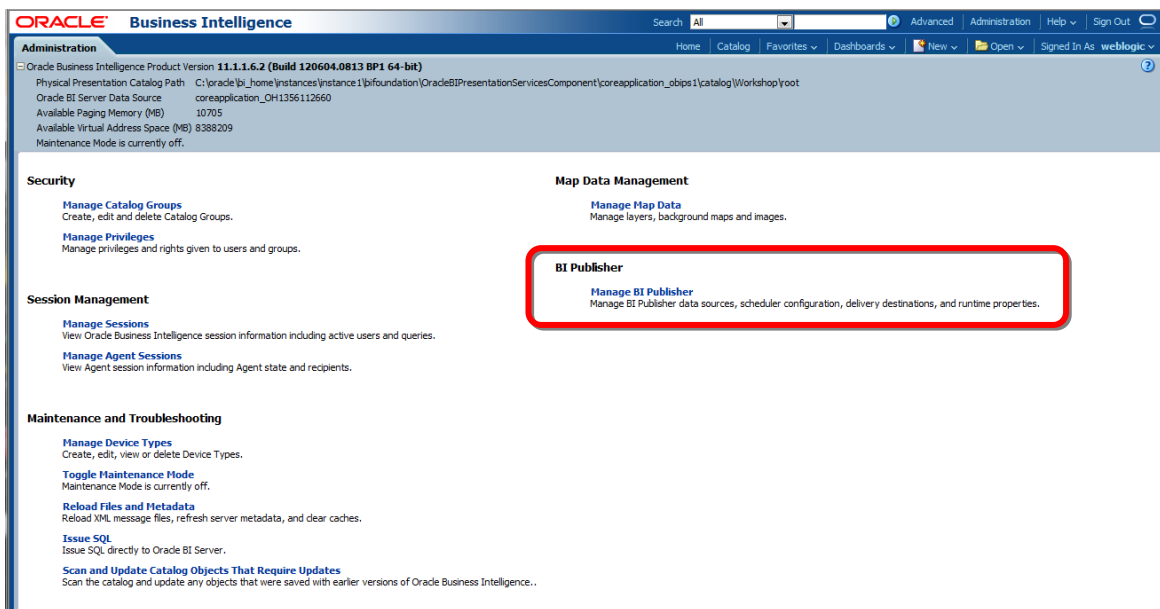
To configure BI Publisher for accessibility, log in to OBIEE as a user that has full administrator privileges, and then select the Administration menu in the top right corner of the main OBIEE user screen.



**Figure 63 – Accessing the Administration Screens**

(If you do not see the “Administration” menu link, your login account does not have the required privileges.)

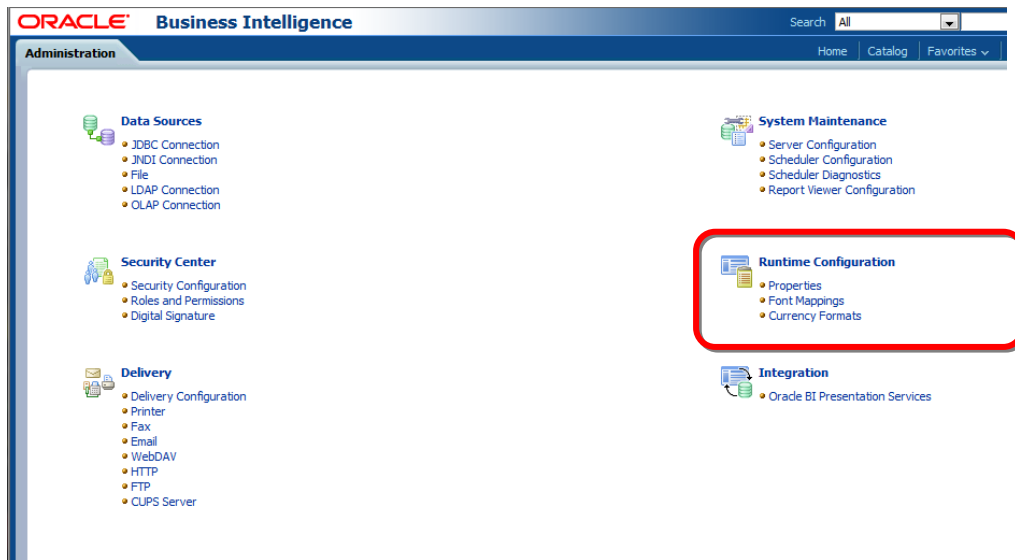
You will now be presented with the general OBIEE administration screen. In the right-hand column, select the “Manage BI Publisher” link.



**Figure 64 – The main OBIEE Administration screen**

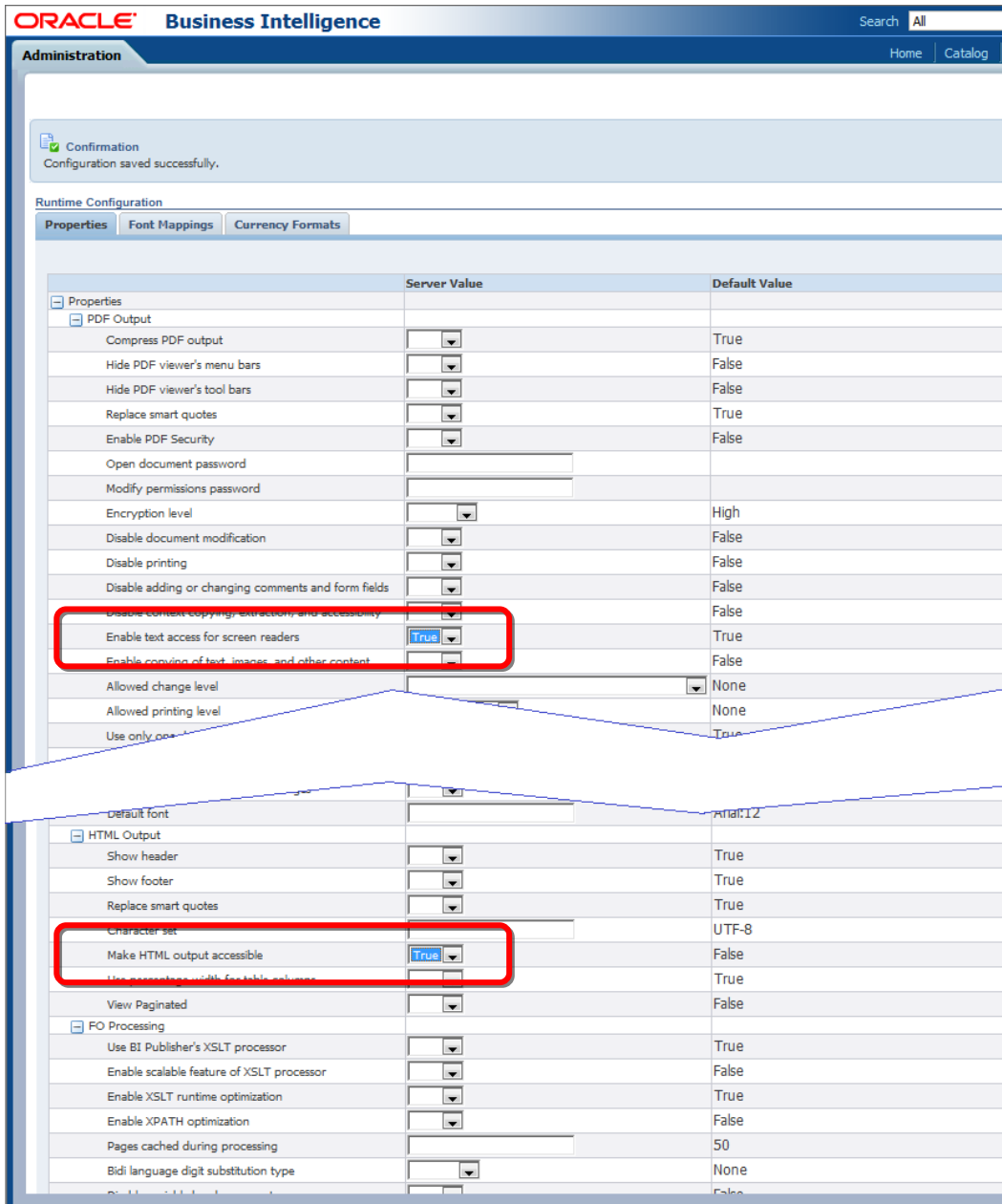
Once you select the Manage BI Publisher link, you will be presented with a screen that lists a number of BI Publisher property categories. From the right-hand column, select the “Properties” link from the Runtime Configuration category.





**Figure 65 – The main BI Publisher Administration screen**

In the Runtime Properties screen that follows, you will need to set two different properties. In the area named “PDF Output”, turn the property labeled “Enable text access for screen readers” to True. Next, scroll down to the HTML Output area. Under the “HTML Output” area, set the “Make HTML output accessible” property to True.



**Figure 66 – Turning on Accessibility Mode for BI Publisher output**

These same runtime configuration settings are also available for each individual report. Any setting changes applied to an individual report will override these general administrative settings.

## BI Publisher: Creating Accessible Content Templates

Naturally, the BI Publisher content creator must still follow all the general guidelines for accessible content listed above (e.g., do not use color alone to convey information, use fonts that magnify well, define ALT text for graphical elements, etc.). These concepts must still be applied to insure that the report output settings are effective in delivering truly accessible content to users.

BI Publisher uses template files to generate the layout of a report. These templates are usually designed as RTF files using MS Word as the template editor. All report designs should apply the following guidelines to ensure that the report content generated by BI Publisher is accessible. (The following information is also listed in the **BI Publisher Best Practices Whitepaper**.)

- Set the document title in the RTF template in the "Title" field in the MS Word document properties dialog:
  - a. For MS Word 2003, use the File | Properties menu selection.
  - b. For MS Word 2007, use the MS Office button, followed by Prepare | Properties.
- For table-oriented data presentations:
  1. Row Headers and Column Headers must both be specified.

Distribution Account	Net Debits	Net Credits
ABC-554-3345-0000	5,654.32	4,530.67
FFF-454-3553-1111	36.40	30.00

**Figure 67 – Including Column and Row Headers in Tables**

2. Define the Table Summary Text by adding a form field with the text "<?table-summary:'Table Summary Text'?>" in the first column of the first row of the table.
3. Repeat the table header on each page by checking the “Repeat as Header Row on the top of each page” for the header row of a table.
  - a. For MS Word 2003, open the Right-Click menu, then select “Properties”.
  - b. For MS Word 2007, open the Right-Click menu, select “Table Properties”, then select the “Row” tab.
4. Add a form field with the text "<?horizontal-break-table:1?>" in the first column of the first row in tables with repeating column headers.

Complicated table structures such as nested tables should be avoided, as well as tables used solely to control field layout. Both of these kinds of table structures can cause problems for screen reader applications.

- For paragraph-oriented information and for data in group break headers, use the correct Word heading style (e.g., "Heading 1", "Heading 2") to have the heading information available in the generated HTML. In MS Word 2007, these format settings are located in the Home ribbon under Styles.
- For images, set the alternative text for images by typing "alt:Picture alternative text" in the "Web" or "Alt Text" tab of the image's properties.
  - a. For MS Word 2003, double-click on the image, or right-click then select "Format Picture", and then select the "Web" tab.
  - b. For MS Word 2007, right-click on the image, select "Size" and then select the "Alt Text" tab.

When using MS Word as your template creator for BI Publisher content, a number of good resources exist that provide information about creating accessible MS Word templates. The Microsoft Office web site offers a free tutorial for Office 2003 that covers accessibility, located at <http://office.microsoft.com/training/training.aspx?AssetID=RC063800961033>. The Adobe web site also provides a set of tips for using MS Word to create accessible output located at [http://blogs.adobe.com/accessibility/assets/WordToPDFReferenceCard\\_v1.pdf](http://blogs.adobe.com/accessibility/assets/WordToPDFReferenceCard_v1.pdf).

---

**Note:** While the Adobe Reference Card specifically discusses creating accessible PDF output, BI Publisher is not coded to generate accessible PDF output. BI Publisher accessibility support still requires the use of HTML output. The Adobe reference card is merely referenced here as a set of useful design techniques.

---

## Additional Enhancements

A number of additional techniques are useful in making your content more accessible. While these items may not be a national accessibility requirement, many states and provinces add to the national requirements. So it is prudent for the content developer to also include these design techniques as well.

### Custom On-Line Help for Accessibility Users

Very few users actually read software documentation. This can be especially problematic for accessibility needs users who may not be familiar with the OBIEE keystroke equivalents documented in this whitepaper.

As a content designer, you can provide custom on-line help in the form of either text objects defined in a report or dashboard, or as dashboard links to HTML documents published into the OBIEE Web Catalog. A variety of techniques are documented here.

### Adding Text Objects to Reports

One of the display objects you can insert into your Analysis designs is an object called a Static Text Object. To add a Static Text object to a report, follow these steps:

1. Launch the Analysis Editor for the selected report.
2. Select the Results tab.
3. Click on the “New View” icon in the toolbar.
4. Scroll down the menu list to “Other Views” to open the sub-menu for alternate view objects.
5. Select the Static Text object.

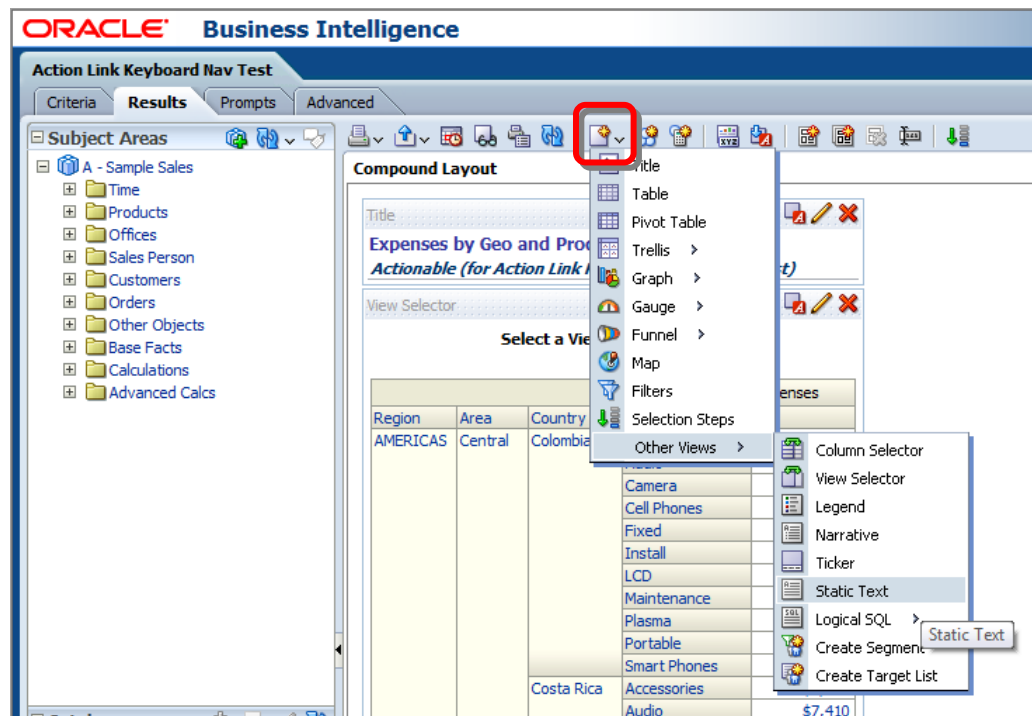
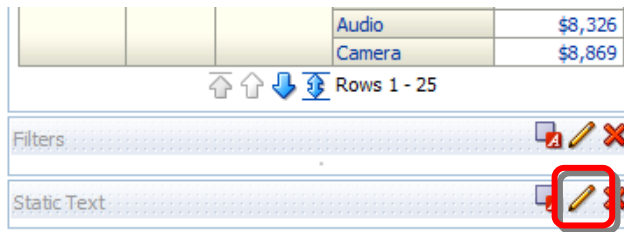


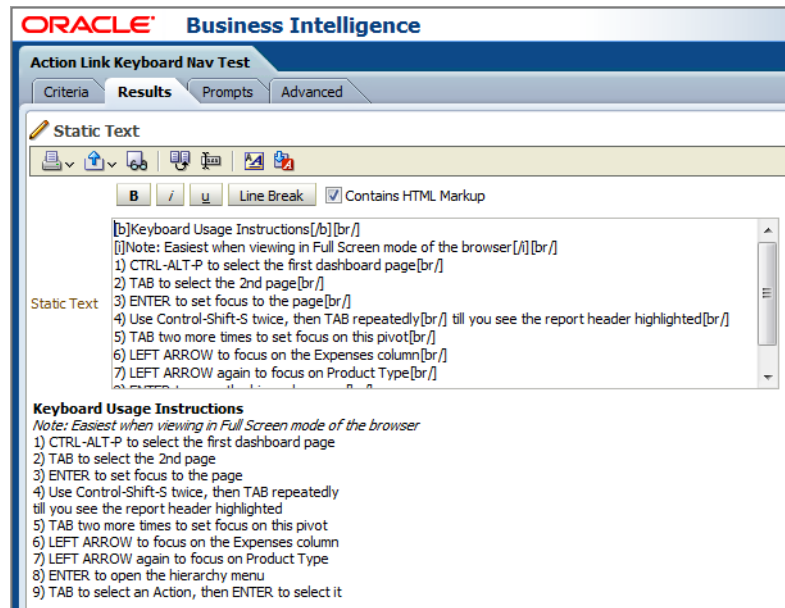
Figure 68 – Adding a Static Text object

6. Enter the Static Text object’s editor by clicking on the Pencil icon in the object’s header.



**Figure 69 – Invoking the Static Text object’s Editor**

7. Enter the desired text. Note that you can use HTML markup tags if you turn on the “Contains HTML Markup” property checkbox.



**Figure 70 – Sample Accessibility Help in a Static Text object**

8. Click on the “Done” button on the top right to save your text and exit the editor.

- In the Results tab, drag the new Static Text into the desired position.

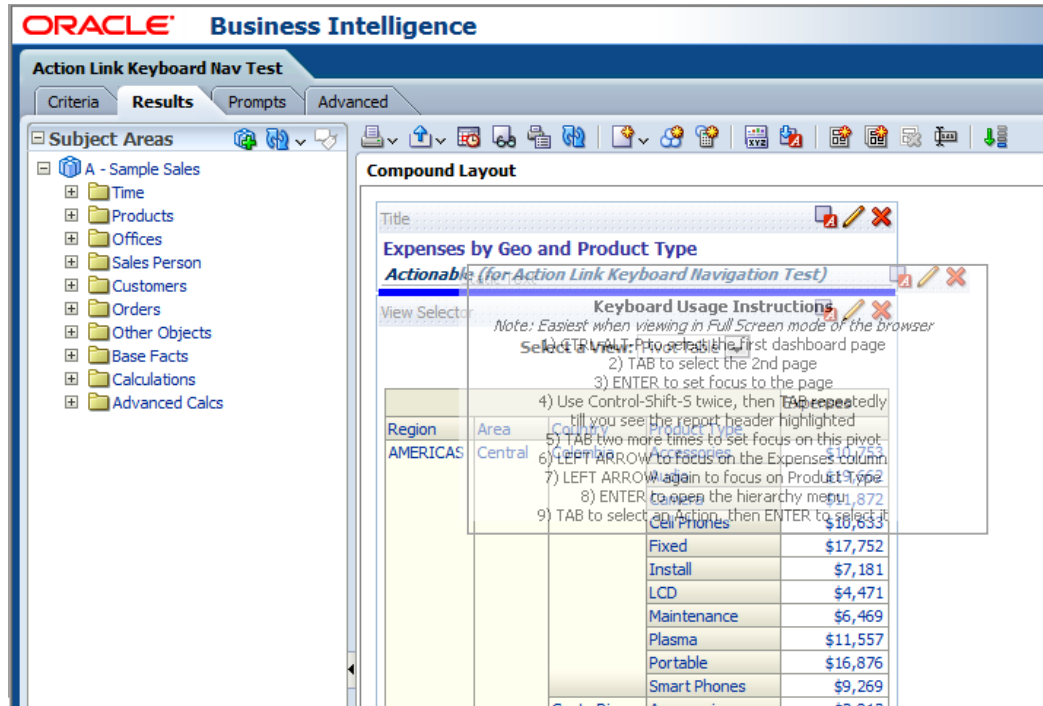


Figure 71 – Positioning the Static Text object

The final analysis report design should look something like this when in Preview mode:

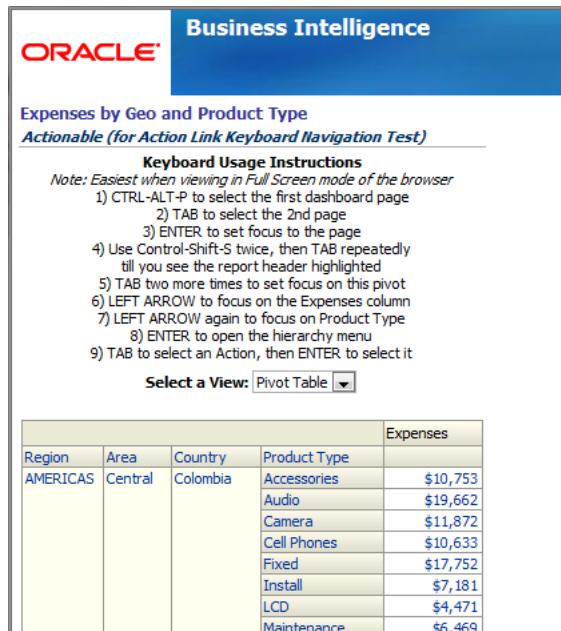
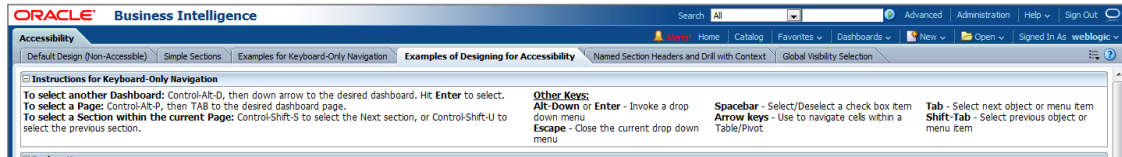


Figure 72 – Final design with static In-line Help text

### Adding Text Objects to Dashboards

If you don't need to add specific In-Line Help text to individual reports, you can provide page-level In-Line Help on your dashboards by adding one or more Text objects to a dashboard page within the Dashboard Editor.

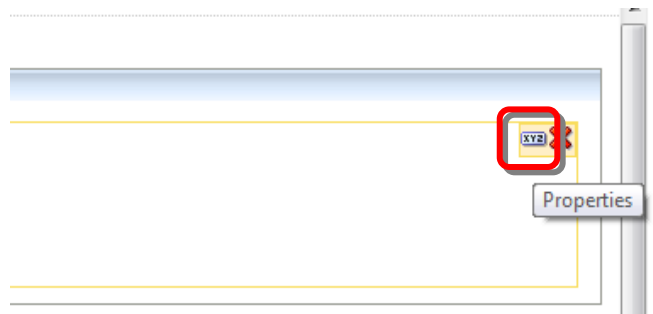
In this example, the goal is to provide a multi-column display of In-Line Help text for keyboard navigation. The final design will look like this:



**Figure 73 – Page-Level In-Line Help**

To add this kind of in-line help text, follow these steps:

1. Select the desired dashboard page and launch the Dashboard Editor.
2. Add a new Section object at the top of the page. Configure the section's properties as required for accessibility to provide an explanatory name and make the section name visible.
3. From the catalog of Dashboard Objects on the left, add a new Text object to the new section.
4. Hover over the upper-right corner of the Text object and select the "Properties" icon.



**Figure 74 – Invoking the Text object's Properties editor**



5. Enter the desired help text. Make sure you turn on the “Contains HTML Markup” checkbox if you want to use HTML tags in your text.

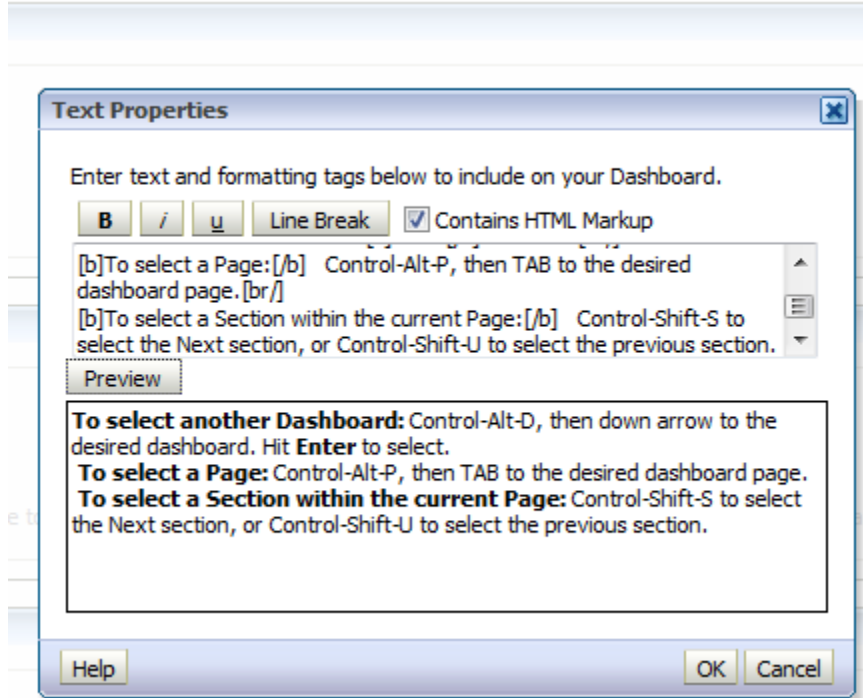


Figure 75 – Defining a Page-Level Text object

6. Add any other Text objects to the same dashboard section to complete the multi-column in-line help layout. Your Dashboard Editor screen should now look something like this:

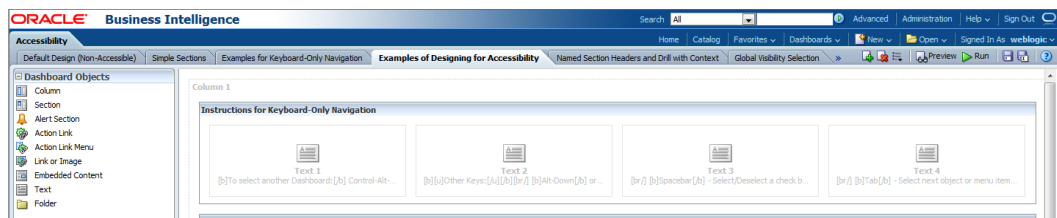


Figure 76 – Dashboard Editor Multi-Column In-Line Help design

7. Save your updated dashboard page design and then run the dashboard. Your display should now look something like what is displayed in Figure 73.

#### Adding Links to On-Line Help documents uploaded into the OBIEE Web Catalog

Most on-line help only needs to be referenced once or twice until the user is familiar with the information. After that, in-line help designs may simply be a waste of valuable screen real-estate.

If your local accessibility standards allow, you may want to use a linked document instead of in-line help text. That way, a user can reference the information when needed, and your dashboard doesn't use up a lot of space that could be displaying other useful content.

The set up for this kind of reference-on-demand custom help requires a few simple steps.

- Create an accessible, HTML-based help document.
- Upload your HTML document to a folder in the OBIEE Web Catalog.
- Add a Link object to your dashboard page that references the custom HTML Help document.

Here are the details for each step:

1. Create an accessible, HTML-based help document. This can be done in any tool that supports creation of HTML files. (In our example, MS Word was used to create an HTML tabular display.)
2. Upload your HTML document to a folder in the OBIEE Web Catalog
  - a. Log into the OBIEE Presentation Server with a user account that has sufficient privileges to create folders and upload documents.
  - b. Select "Catalog" from the main OBIEE menu and navigate to the desired target folder location where you want to store your custom help document.
  - c. Click on the "Upload" link in the Tasks list and upload your custom help file.

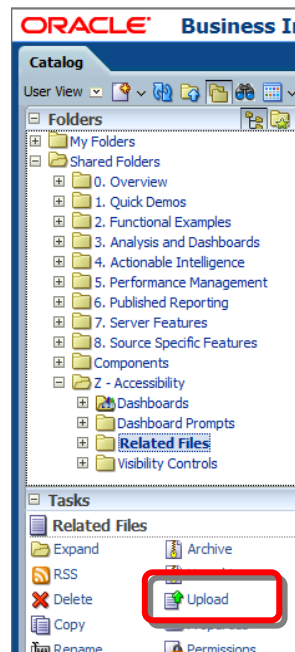


Figure 77 – Selecting the "Upload a File to the current folder" Task

- d. Use the “Browse” button in the dialog box that appears to select your desired file.
  - e. Once uploaded, click on the “Download” link to display your custom Help document in your browser. Copy the URL to the Windows clipboard in preparation for the next step.
  - f. Click the browser’s “Back” button to return to OBIEE.
3. Add a Link object to your dashboard page that references the custom HTML Help document.
    - a. Select the desired dashboard page and launch the Dashboard Editor.
    - b. From the catalog of Dashboard objects on the left, add a “Link or Image” object to a section in your dashboard page.
    - c. Click on the “Properties” icon to invoke the Link object editor.
    - d. Define a Caption and set the Target to be “New Window”. Set the Destination to be “URL” and then paste in the help document’s URL that you copied to the clipboard at the end of Step 2. Do NOT use the Browse button to select the document, as it only allows selection of report objects.
    - e. Edit the pasted URL to remove the reference to your specific server and port. (In other words, your link’s URL should start with “/analytics”.) This will make it a relative-address URL that will be portable to other server installations, such as when moving from Development to Test to Production.

Your final Properties dialog should look something like this:

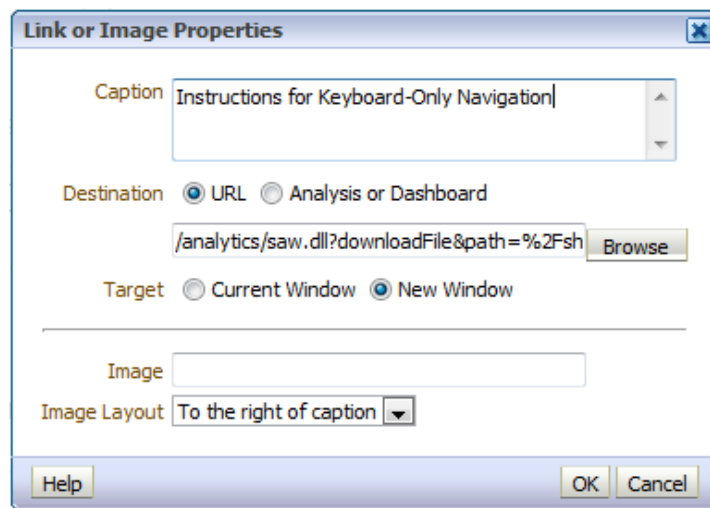


Figure 78 – Link properties for custom help document

Specifically, in this example, the details are:

Original Full URL	<a href="http://192.168.56.101:7001/analytics/saw.dll?downloadFile&amp;path=%2Fshared%2FZ%20-%20Accessibility%2FRelated%20Files%2FKeyboard_Navigation.htm">http://192.168.56.101:7001/analytics/saw.dll?downloadFile&amp;path=%2Fshared%2FZ%20-%20Accessibility%2FRelated%20Files%2FKeyboard_Navigation.htm</a>
Final URL Entry in Properties	<a href="/analytics/saw.dll?downloadFile&amp;path=%2Fshared%2FZ%20-%20Accessibility%2FRelated%20Files%2FKeyboard_Navigation.htm">/analytics/saw.dll?downloadFile&amp;path=%2Fshared%2FZ%20-%20Accessibility%2FRelated%20Files%2FKeyboard_Navigation.htm</a>

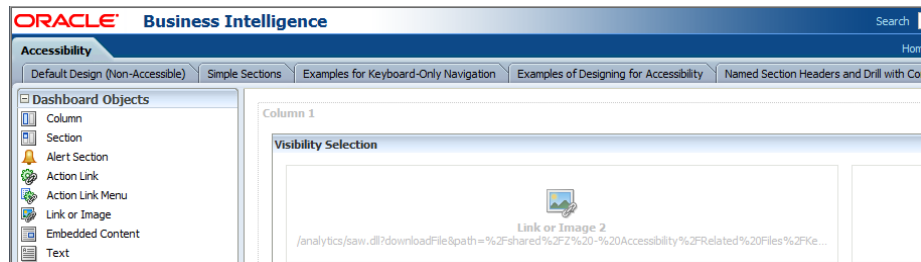
The “http://<serveraddress:port>” portion of the original URL is to be removed.

---

**Note:** The removal of the server:port reference converts the URL reference to be a relative address, allowing administrators to migrate content from Development to Test to Production without having to change the specific server reference used.

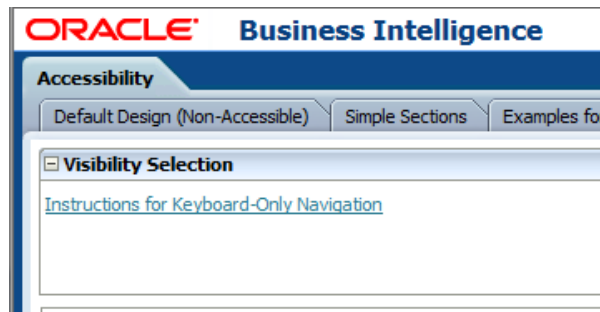
---

- f. Click the OK button to save the link object properties. Your Dashboard Editor screen should now look like this:



**Figure 79 – Dashboard Editor layout with Link to Custom Help**

- g. Save the dashboard and run it and it should look something like this:



**Figure 80 – Link to Custom Help**

- h. Click on the “Instructions” link and a separate browser window should open, displaying your custom help.

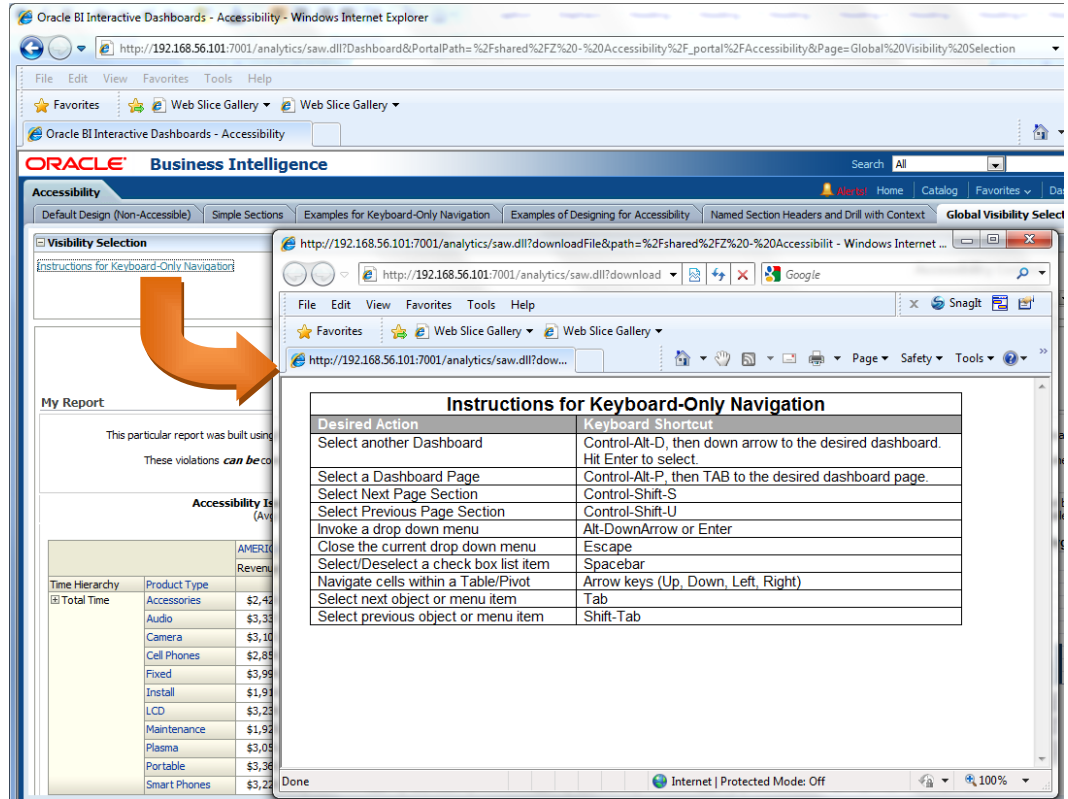


Figure 81 – Link to Custom Help in Action

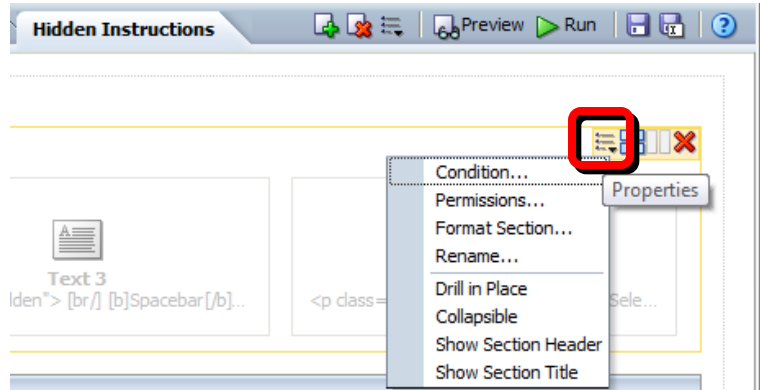
### Adding “Hidden Text” for Screen Reader Users

While the above technique of adding a link to a dashboard page is a more efficient use of screen space than adding text elements, it still does take up some space. Sometimes you need to provide additional explanatory text to users of screen readers that you don’t want to take up any screen space for other users. In fact, you may not want sighted users to know that any additional information is available at all. Screen readers are smart, though. If you merely set the text to be the same color as the background, the screen reader will note this and usually ignore the text.

You can accomplish this goal of hiding text that *will* be read by screen readers by adding a text style to your narratives that places the text display off-screen. This technique still allows a screen reader user to hear this extra text spoken, but sighted users will not see it, nor will it take up any screen space for sighted users.

To implement this solution, start by adding the same text objects to your dashboard as shown above on page 71, “Adding Text Objects to Dashboards”. Then, follow the additional steps below to set up the section and text to be visually hidden but available to screen reader tools.

- 1) While still in the Dashboard Editor, click on the “Properties” icon of the section containing your text objects to invoke its Properties menu.



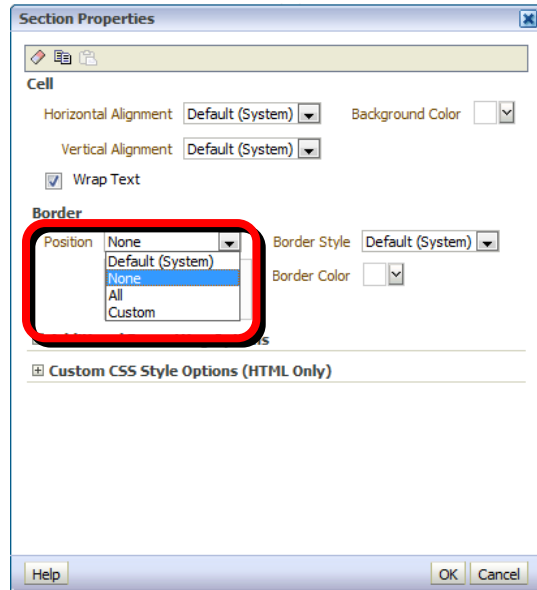
**Figure 82 – Invoking the Section Properties Menu**

- 2) Select the “Rename...” menu item and enter the following text in the dialog box that appears:

**Hit the Down Arrow key to read Instructions for Keyboard-Only Navigation**

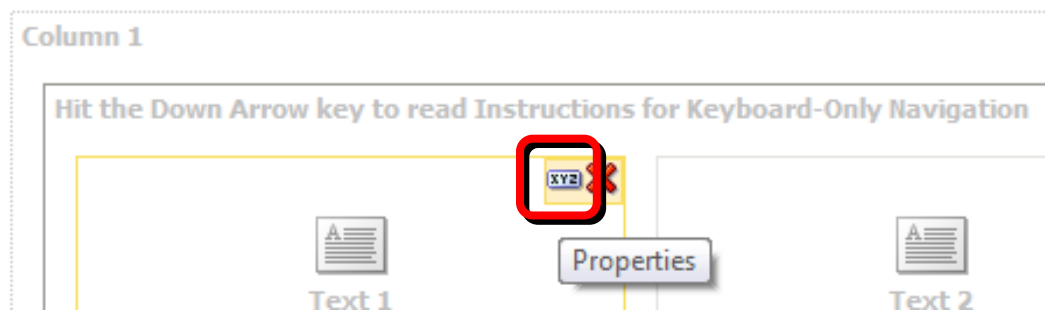
You include the information about the down arrow in the section name so that your user’s screen reader will speak the appropriate information about how to navigate into the hidden text when they land on the “hidden text” section icon.

- 3) Invoke the Section Properties menu a second time and select “Format Section...”, then set the Border Position to “None” in order to hide it.



**Figure 83 – Hiding the Section Border**

- 4) Invoke the Section Properties menu again and make sure you turn OFF the “Collapsible”, “Show Section Header” and “Show Section Title” options. Turning off these options along with setting the Border Position to “None” will ensure that non-Accessibility Mode users will not see anything on the screen to indicate that the section even exists.
- 5) Select the first (ie, left-most) text object and invoke its Properties menu.



**Figure 84 – Invoking the Text 1 object's Properties dialog box**

- 6) In the dialog box which appears, make sure that the “Contains HTML Markup” checkbox is turned ON, and then add the following HTML code to the top of the text box:

```
<STYLE>
.hiddenText
{position:absolute;
left:-10000px;
top:auto;
width:1px;
height:1px;
overflow:hidden;}
</STYLE>
```

This establishes an HTML style named hiddenText that can be used throughout the dashboard page to place text 10,000 pixels to the left of the left side of the viewable screen.

- 7) Immediately after the end of the style definition, frame the regular text in your object with the following:

```
<p class="hiddenText">
Regular text goes here
</p>
```

- 8) In the case of the example Text 1 object, the entire final text should read as follows:

```
<STYLE>
.hiddenText
{position:absolute;
left:-1000px;
top:auto;
width:1px;
height:1px;
overflow:hidden;}
</STYLE>
<p class="hiddenText">
<strong>To select another Dashboard:</strong> Use Control-Alt-D,
then down arrow or Tab to the desired dashboard link. Hit
<strong>Enter</strong> to select.<br>
<strong>To select a Page:</strong> Use Control-Alt-P, then TAB to
the desired dashboard page.<br>
<strong>To select a Section within the current Page:</strong> Use
Control-Shift-S to select the Next section, or Control-Shift-U to select the
previous section.
</p>
```

- 9) Open the Properties dialog of all other text objects meant to be used for hidden text, and frame their text in the same paragraph class setting code as shown in Step 5. Also, make **sure** you have the “Contains HTML Markup” checkbox turned ON in each case.
- 10) Your final dashboard design should look something like this inside the Dashboard Editor:

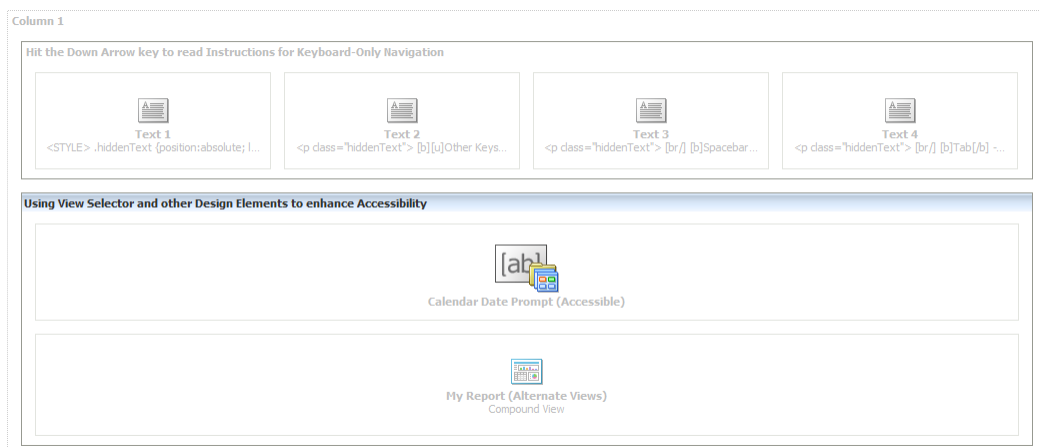


Figure 85 – Final "Hidden Text" Design

- 11) Save the dashboard design and run it.



When running in Accessibility Mode, the user can easily navigate to the hidden text section using the Next/Previous Section keyboard shortcuts. Their screen reader will then tell them to use the Down Arrow key to read the additional hidden text. Each press of the down arrow key will read the next line of instructions. When they get to the last line of hidden instructions, the next down arrow key press will take them to the next section on the dashboard.

Sighted users will actually see nothing out of the ordinary. Notice that, in regular display mode, no indication of the hidden text section appears at all.

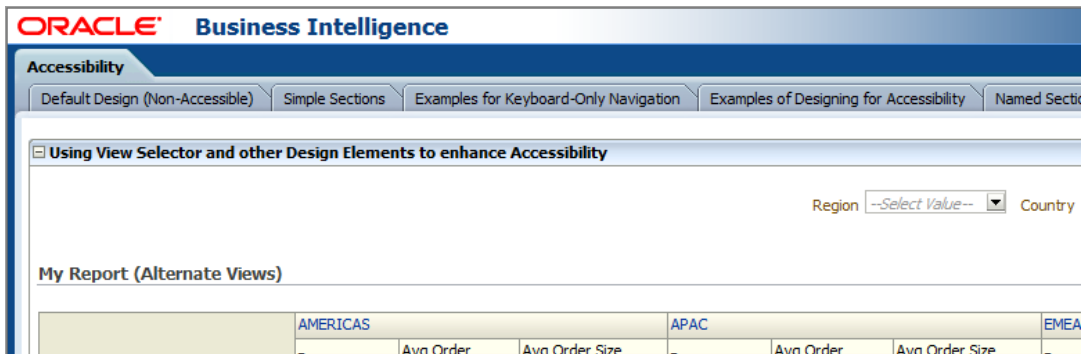


Figure 86 – Hidden Text in Regular Mode

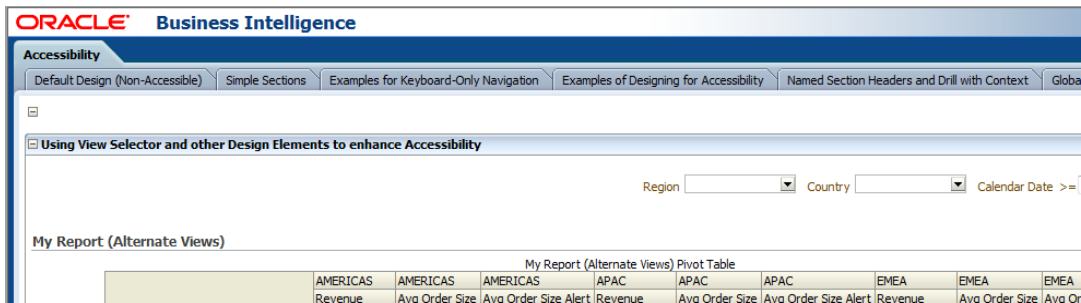


Figure 87 – Hidden Text in Accessibility Mode

When Accessibility Mode is turned on, screen real-estate is not really an issue since most users of this mode are not sighted users. However, even in Accessibility Mode, the necessary display of the section’s expand/collapse widget for navigation support takes a minimal amount of space at the top of the screen.

---

**Note:** Since this technique specifically turns off the display of the Section Header and Title, this Hidden Text Section will *not* be listed by the screen reader’s “List Headings” keyboard shortcut hotkey. Screen reader users will have to use the OBIEE Next/Previous Section hotkeys or tab into it instead.

---

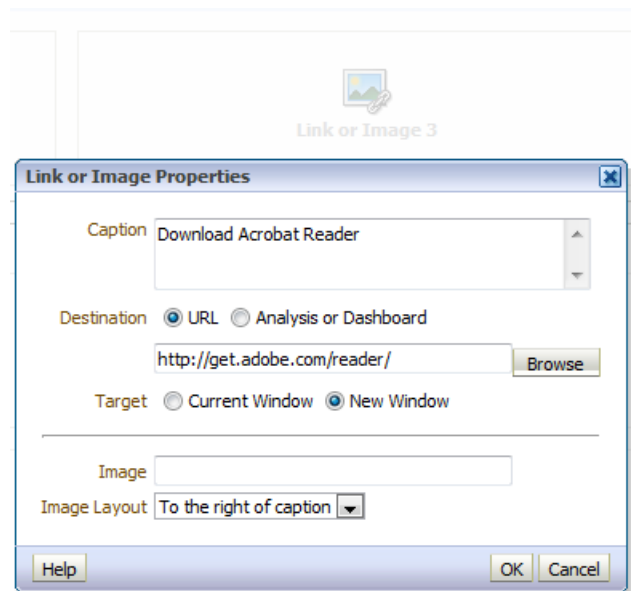
### Links to Required Plug-Ins

Some localities require that a link be present on your specific web site for installing any required browser plug-ins (even ones that nearly everyone already has installed). For OBIEE plug-ins, links already exist on the OBIEE Home Page, as previously shown in Figure 6 on page 17.

For third-party plug-ins such as Adobe Acrobat Reader, you will need to add a custom link to your dashboard page yourself. To do this, simply repeat the actions you used to create a Link object to your custom help document. In this case, the steps would be modified as follows:

1. Navigate on the Internet to the location where the 3<sup>rd</sup> party plug-in can be downloaded.
2. Copy the URL of this location to the Windows Clipboard..
3. Add a Link object to your dashboard page that references the download URL location you copied in step 2.

The next figure illustrates an example of this using the location for downloading the Adobe Acrobat Reader, which is <http://get.adobe.com/reader/>.



**Figure 88 – Defining a Link object to download a 3rd party plug-in**

The end result of this Link object definition would look something like this:

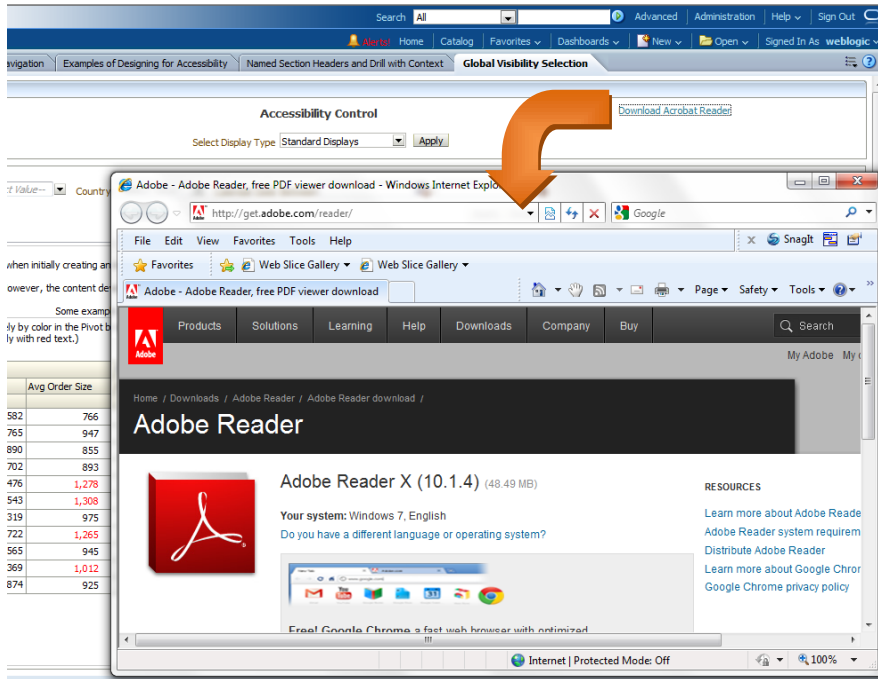


Figure 89 – Link to download 3rd Party Plug-in in Action

### Controlling Visibility of Custom Links

The v11.1.1.6 release of OBIEE added the ability to define custom URL links in the OBIEE global header area. This is done by adding the link definitions to a customlinks.xml file. Details of how to do this can be found in the “Configuring and Managing Analyses and Dashboards” chapter of the OBIEE System Administrator’s Guide. Look in the section titled “Manually Changing Presentation Settings”.

For deployments that require support for users with accessibility needs, you need to make sure that the links you want to remain available when in Accessibility Mode are tagged with the optional “vpat=true” property.

link: vpat	Optional	Boolean	Specifies that in accessibility mode, the link is available only when the vpat attribute is set to true. Values are true and false.
------------	----------	---------	---

For example, the following link definition will be available when the user is running in Accessibility Mode:

```
<link id="12" name="Oracle 1" description="Oracle" src="http://www.oracle.com" target="blank" vpat="true" >
```

## Conclusion

The Oracle Business Intelligence Suite Enterprise Edition 11g provides a variety of features to support creation and deployment of accessible content to users with disabilities. These features must be leveraged properly in accordance with the guidelines discussed in this whitepaper to insure equivalent access to these users.

For further details, please refer to the information found on the Oracle web site at <http://www.oracle.com/accessibility/index.html> and the specific product documentation sets:

- For Oracle Dashboards and BI Publisher:  
[http://www.oracle.com/technology/documentation/bi\\_ee.html](http://www.oracle.com/technology/documentation/bi_ee.html)
- For Interactive Reporting and the EPM Workspace:  
<http://www.oracle.com/technology/documentation/epm.html>



A Guide to Accessible Content for Oracle  
Business Intelligence Suite 11g  
October 2012

Author: Mark Ostroff  
Contributors: Tim Dexter, Don Mauck,  
Peter Wallack

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
oracle.com



| Oracle is committed to developing practices and products that help protect the environment

Copyright © 2012, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.