

A Guide to Accessible Content for Oracle Business Intelligence Suite Enterprise Edition Plus

An Oracle White Paper
December 2009

Introduction	1
For Those New to the Issue of Accessibility.....	2
Starting Resources	2
For End Users	2
Keyboard Navigation	2
Notes for Screen Reading Software.....	5
BI Publisher: Turning on Accessibility Mode for the UI.....	6
IR & EPM Workspace: End User selection of Accessibility Mode ..	8
For Developers.....	9
Common Misconceptions	9
General Guidelines for Creating Accessible Content	9
Dashboard-Specific Guidelines.....	11
OBIEE-Specific Techniques for Implementing Accessibility	13
Oracle Dashboards: Turning on Accessibility Mode.....	13
Oracle Dashboards: Optional Accessibility Settings.....	14
Oracle Dashboards: Specific Content Techniques.....	14
BI Publisher: Turning on Accessibility Mode for Output.....	27
BI Publisher: Creating Accessible Content Templates	28
IR & EPM Workspace: Turning on Accessibility Mode	30
Accessible Ad-Hoc Queries: Setting up for Casual Users.....	33
Accessible Ad-Hoc Queries: Setting up for Power Users	33
Accessible Scheduling: Setting up the EPM Workspace.....	36
Conclusion	37

Introduction

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

The accessibility features in the OBIEE+ 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+ accessibility features fall into three general categories:

- 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 3rd party assistive technology products.
- Content design capabilities that make it possible for content creators to build BI content that supports users with accessibility needs.
- Keyboard shortcuts that make it easier to navigate content for users with limited or no ability to use a mouse.

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](http://www.access-board.gov/sec508/update-index.htm)
<http://www.access-board.gov/sec508/update-index.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/WCAG/)
<http://www.w3.org/TR/WCAG/>
- [Roadmap for Accessible Rich Internet Applications \(WAI-ARIA\)](http://www.w3.org/TR/aria-roadmap/)
<http://www.w3.org/TR/aria-roadmap/>

For End Users

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

Keyboard Navigation

A variety of keyboard equivalents are provided by Oracle Dashboards, BI Publisher, EPM Workspace, Interactive Reporting and Smart View.

For the keyboard information for the EPM System 11 components, please refer to Appendix A, Accessibility in the EPM Workspace User's Guide located on the Oracle Technology Network at http://download.oracle.com/docs/cd/E12825_01/epm.111/bpmui_user.pdf.

The available keyboard commands for Oracle Dashboards and BI Publisher are listed here, grouped based on module, menu and control as listed below.

Keyboard Accelerators 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.

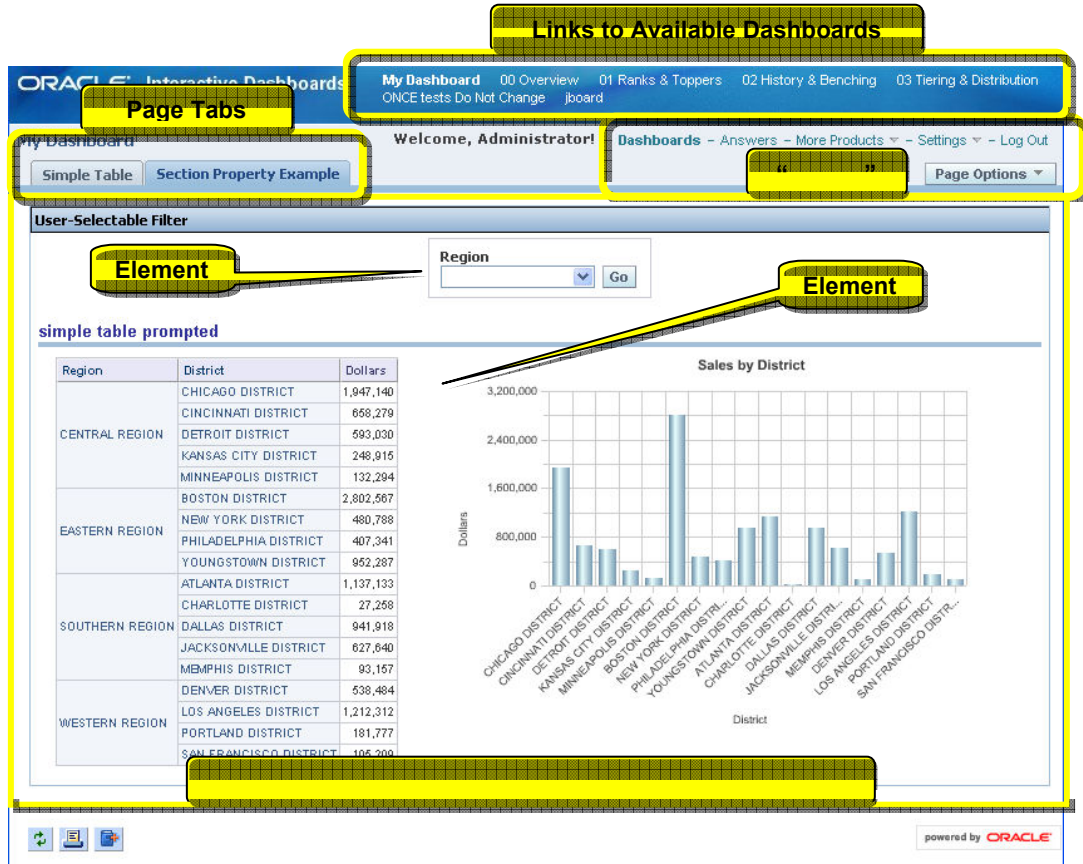


Figure 1 - The different regions of an Oracle Dashboard screen

At the top of the screen, in what is known as the banner region, are a series of links that are associated with the various dashboards that are available to the user. Each user may see a different set of dashboard links, but the first dashboard link for every user is usually associated with the personal dashboard named “My Dashboard”.

The area immediately below the banner 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.

On the right side of the dashboard header is the menu area. Items in the menu area allow you to select functionality associated with other OBIEE+ modules, 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) and to log out of the system.

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 can optionally be collapsed to temporarily hide their content.

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

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

Action	Keyboard Shortcut
Focus on the first link of Available Dashboards	Ctrl+Shift+F2
Navigate to the next Dashboard link	Tab (when focused on a link)
Select the Dashboard whose link has the current focus	Enter
Focus on the first dashboard page tab on the currently selected dashboard (applies to multi-page dashboards only)	Ctrl+Shift+F3
Navigate to the next Dashboard page tab	Tab (when focused on a tab)
Select the dashboard page whose tab currently has focus	Enter
Move backward through sections on dashboard page	Ctrl+Shift+U
Move forward through sections on dashboard page	Ctrl+Shift+S
Move backward through elements of section	Ctrl+Shift+F7
Move forward through elements of a section	Ctrl+Shift+F8
Set focus to Table Object (to activate screen reader annotations)	Tab, when screen reader announces a table

NOTE: Users of the Firefox browser and some versions of Internet Explorer may not automatically load the table annotations needed by screen reader software. JAWS users will need to reload the screen reader's buffer. You do this by pressing Insert+Esc after you press Tab to set focus on the Table.

Keyboard Navigation of Menus in Oracle Dashboards and BI Publisher

Action	Keyboard Shortcut
Open Menu	Enter
Move Up/Down menu or submenu items	Up/Down Arrow Key
Execute Menu Item action	Enter
Open Submenu	Enter
Close menu or submenu	Esc

TIP: To quickly select the menu area, you can use Ctrl+Shift+F3 to select the first page tab, and then use Shift-Tab to navigate back into the menu area (landing on the “Logout” menu item).

Keyboard Navigation of Drop Downs and Combo Boxes in Oracle Dashboards and BI Publisher

Action	Keyboard Shortcut
Opens drop down and combo boxes	Alt+Up/Down Arrow Key
Shows next/previous item in a combo box	Up/Down Arrow Key
Selects option	Enter
Activate selected option	Tab to “GO” button, then Enter

Notes for Screen Reading Software

When navigating an Oracle dashboard, the virtual PC Cursor mode on your screen reader can be either on or off. However, when navigating within table objects on the dashboard, the virtual PC Cursor should be turned on. (For example: Insert+Z when using JAWS.)

Keyboard Navigation within a Table

In data tables and pivots, in order for assistive technology to work, each data cell must be associated with the corresponding column and row headers through annotation. To improve overall system performance, this annotation is not added when the Dashboard is initially loaded. Only when a table receives focus is the annotation generated.

To navigate the cells within a table, ensure that virtual PC Cursor Mode is turned on. You should then use the screen reader table navigation keyboard shortcuts instead of tabbing, as not every table cell can be tabbed into. For instance, if using JAWS, Alt+Ctrl+Right arrow will take you to the next cell on the right. Please refer to your assistive technology documentation for all of the applicable table navigation shortcuts.

Example of navigating through a dashboard using the JAWS screen reader

The following example illustrates how a JAWS user would navigate within an Oracle dashboard.

1. Navigate to the desired dashboard:
 - a. Press Ctrl+Shift+F3 to move the focus to the first dashboard link.
 - b. Hit TAB until you focus on the desired dashboard, then hit Enter.
2. Navigate to the desired page on the dashboard
 - a. Press Ctrl+Shift+F2 to move the focus to the page tabs.
 - b. Hit TAB until focus moves to the desired page, then press Enter.
3. Use the TAB key or Ctrl+Shift+S to move through sections of a dashboard. You may use TAB or Ctrl+Shift+F7 to move through different elements of a section.
4. Once you bring focus to a pivot or table, hit TAB. If you are using the Firefox browser, then hit Insert+Esc to refresh the virtual screen buffer to access the table annotations.
5. Make sure the Virtual PC Cursor mode is on (Insert+Z in JAWS). You will then be able to use the Table navigation keys documented in your assistive technology.

BI Publisher: Turning on Accessibility Mode for the UI

The BI Publisher module has its own user preferences settings, apart from OBIEE+ itself. The Accessibility Mode for the BI Publisher user interface is set for each individual user. (The OBIEE+ Accessibility Mode for Oracle Dashboards is turned on by an administrator via a global configuration setting discussed later and is not available to end users.)

Users can turn on Accessibility Mode in BI Publisher themselves by selecting the “Preferences” link from the main BI Publisher screen.

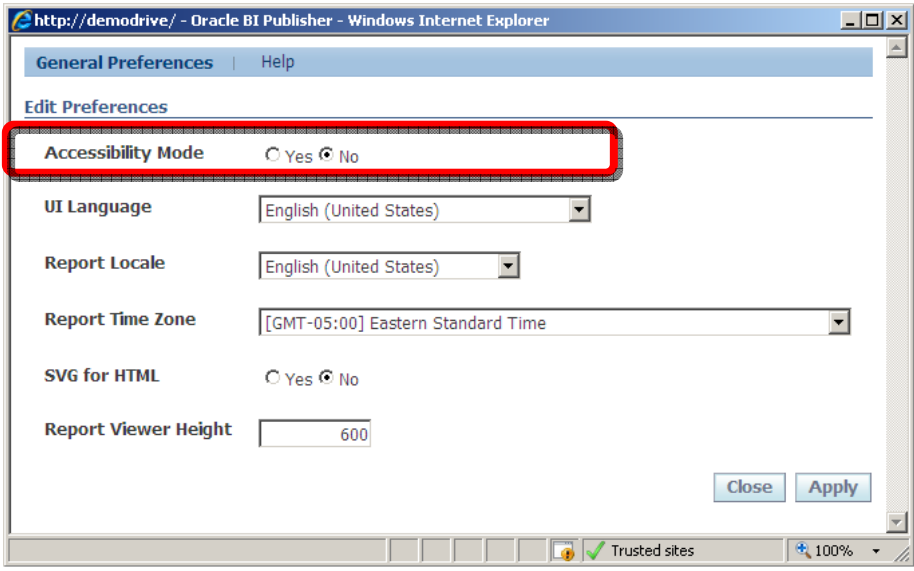


Figure 2 - BI Publisher User Preferences

The first entry on the BI Publisher’s Preferences screen allows you to turn on accessibility mode by selecting the “Yes” radio button. Selecting this option will render the BI Publisher folders in an accessible tree structure that supports keyboard-based navigation.

NOTE: When setting this mode from the user’s Preferences screen, you must select “Apply”, then log out entirely and log back in for the system to recognize this change.

If using BI Publisher in stand-alone mode, this setting is also available on the login screen as a checkbox.

IR & EPM Workspace: End User selection of Accessibility Mode

You can turn on Accessibility Mode in the User Preferences screen in the EPM Workspace. If you have low visual acuity but do not rely on a screen reader, you can select a high-contrast display mode on the same screen.

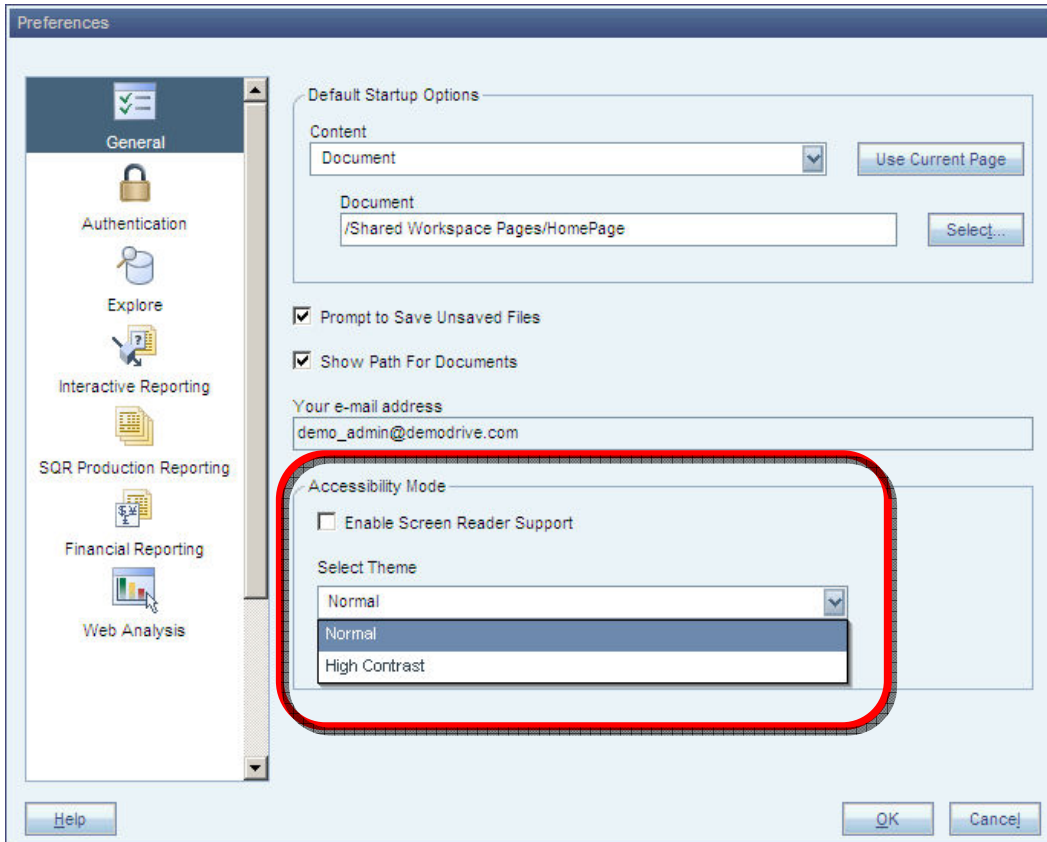


Figure 3 - EPM System 11 Accessibility Settings

From the main EPM Workspace screen, select File | User Preferences from the main menu. The General Preferences screen will then appear. In the lower right portion of the screen, you can turn on screen reader support and/or select a high contrast display theme. Turning on screen reader support tells the system to provide the additional HTML markup those assistive technologies need.

NOTE: You will have select “OK”, then log out, close your browser entirely and log back in for the system to recognize any changes.

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. As a result, they may erroneously report accessible elements as non-accessible. Therefore, accessibility is ultimately the responsibility of the content creator. 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.

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

Different types of charts require different design elements to insure that they are accessible. The default settings for most chart types will not create accessible charts automatically. 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
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.

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 screen reader users will be able to access it without having to navigate the entire screen.

Provide sufficient Textual Information

Graphics can not 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.

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 (such as a “stock ticker” display widget). Make sure that you are familiar with all legally mandated design prohibitions that apply in your locality as well and avoid using those elements.

OBIEE-Specific 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+ should also be employed. The full OBIEE+ product suite consists of a number of modules, each of which covers different aspects of accessibility. Therefore, a completely accessible solution requires several different configuration steps. In addition, different modules employ different specific techniques to create accessible content.

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

- Oracle Dashboards – for content consumption and casual user ad-hoc queries
- BI Publisher – for production (canned) reporting
- Hyperion Interactive Reporting – for ad-hoc querying by power users
- EPM Workspace – for scheduling

The Hyperion Interactive Reporting module (IR) is accessed from within the EPM Workspace, so configuring accessibility for the EPM Workspace will also cover the UI portion of Interactive Reporting. There are, however, a few additional steps to configure IR and the EPM Workspace for use as the accessible ad-hoc query and scheduling tools, which are noted below.

Oracle Dashboards: Turning on Accessibility Mode

In order for Oracle dashboards to be read by assistive technologies, you must first enable the Accessibility mode. To do this, you need to edit the file `instanceconfig.xml`. Open this file in edit mode and add the line:

```
<Enable508>true</Enable508>
```

Adding this tag allows assistive technologies to read dashboards. It enables tables and pivots to be annotated with the necessary information for screen readers and flattens the menu items in report links to become simple HTML links that are easier to navigate.

The `instanceconfig.xml` file is typically found in the `OracleBIData\web\config` directory.

NOTE: Any changes made to the instanceconfig.xml file are global and will affect all users. You will have to stop and restart the Oracle Presentation Server for the system to recognize any changes made to the instanceconfig.xml file.

Oracle Dashboards: Optional Accessibility Settings

Changing the following settings in the instanceconfig.xml file is not required. However, they may be helpful to make dashboard navigation more accessible.

The DashboardMaxBeforeMenu Setting

When more than 15 dashboard names are available for display, the default behavior of Oracle Dashboards is to create a drop-down list of the dashboards from which users can choose. By modifying the instanceconfig.xml file, you can change the number of dashboard names to display as links before the presentation is changed to a drop-down list. This modification can make it easier for users of assistive technology to navigate the dashboard list.

Modify the following entry in the instanceconfig.xml file. The minimum value is 1.

```
<DashboardMaxBeforeMenu>1000</DashboardMaxBeforeMenu>
```

The DashboardInlineLoad Setting

By default, Oracle dashboards only refresh those portions of the screen that need to be updated when a user changes a filter setting or selects a display option such as a View Selector. This feature is also known as Partial Page Refresh and is designed to provide better overall display performance. By turning off the DashboardInlineLoad feature, the entire dashboard page will be refreshed upon any action that requires any portion of the screen to be updated. Forcing a complete page refresh can facilitate screen reading by assistive technology.

To force a full page refresh on any updates, modify the following entry in the instanceconfig.xml file as shown below. The default value is true.

```
<DashboardInlineLoad>false</DashboardInlineLoad>
```

Oracle Dashboards: Specific Content Techniques

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

Screen Organization

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

For example, screen readers assume a logical screen flow from left-to-right and top-to-bottom. They also have no ability to tell if screen content they have already scanned has changed since it was read, unless the entire page is refreshed. Therefore if the `DashboardInlineLoad` setting is set to true, then dashboard designers need to place any controls that might change the screen's content on the screen before the content that would be affected by the control.

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 "Previous Section" keyboard shortcut 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.

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 table" and "test link") 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.

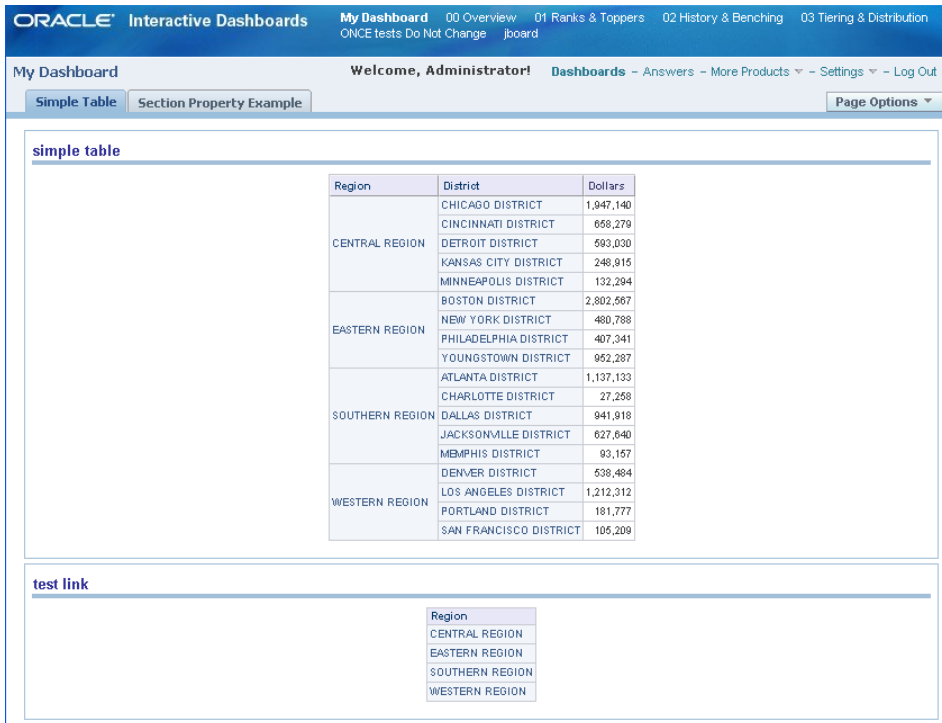


Figure 4 - Default display of a multi-section dashboard

To define section names, select “Page Options” and choose “Edit Dashboard”. Once in the Dashboard Editor, select the “Rename” button on top of the section object. In the dialog box that appears, type in a name that describes the contents of that section and turn on the “Display Section Heading” checkbox. Repeat this process for each section on the dashboard.

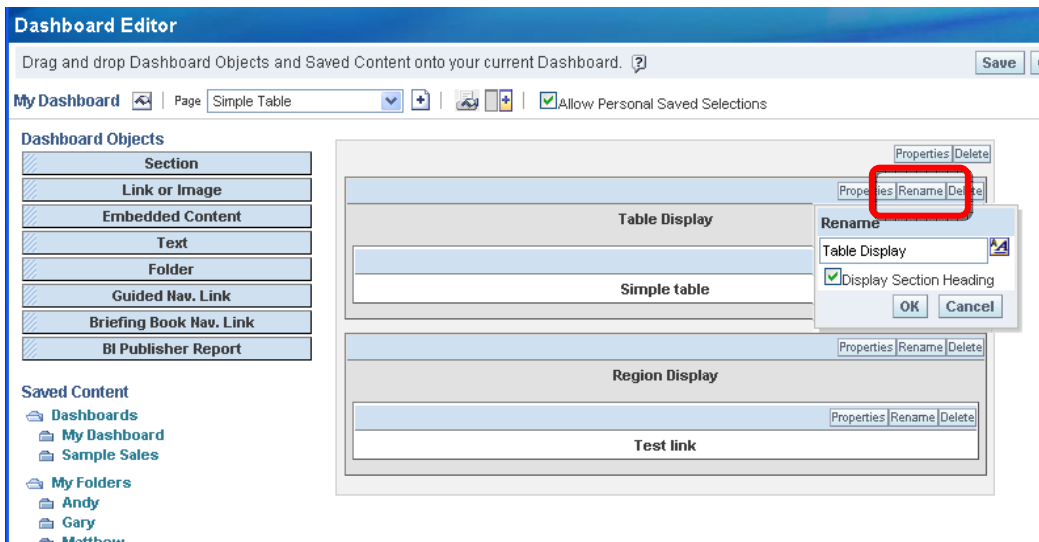


Figure 5 - Defining a visible Section Name

When screen readers announce section names, Oracle Dashboards will automatically add the word “Section” in front of the name you define. So avoid using “section” as part of the name.

The resulting dashboard now will look like this.

The screenshot shows an Oracle Interactive Dashboard interface. At the top, there is a navigation bar with the Oracle logo and 'Interactive Dashboards' text. Below this, a secondary navigation bar contains 'My Dashboard' and several menu items: '00 Overview', '01 Ranks & Toppers', '02 History & Benching', and '03 Tiering & Distribution'. A welcome message 'Welcome, Administrator!' is displayed, along with 'Dashboards' and a list of links: 'Answers', 'More Products', 'Settings', and 'Log Out'. Below the navigation, there are two tabs: 'Simple Table' (selected) and 'Section Property Example'. A 'Page Options' dropdown menu is also visible. The main content area is divided into two sections. The first section, titled 'Table Display', contains a sub-section 'simple table' which displays a table with three columns: 'Region', 'District', and 'Dollars'. The table lists data for four regions: Central, Eastern, Southern, and Western. The second section, titled 'Region Display', contains a sub-section 'test link' which displays a list of region names: 'CENTRAL REGION', 'EASTERN REGION', 'SOUTHERN REGION', and 'WESTERN REGION'.

Region	District	Dollars
CENTRAL REGION	CHICAGO DISTRICT	1,947,140
	CINCINNATI DISTRICT	658,279
	DETROIT DISTRICT	593,030
	KANSAS CITY DISTRICT	248,915
	MINNEAPOLIS DISTRICT	132,294
EASTERN REGION	BOSTON DISTRICT	2,802,567
	NEW YORK DISTRICT	480,788
	PHILADELPHIA DISTRICT	407,341
SOUTHERN REGION	YOUNGSTOWN DISTRICT	952,287
	ATLANTA DISTRICT	1,137,133
	CHARLOTTE DISTRICT	27,258
	DALLAS DISTRICT	941,918
WESTERN REGION	JACKSONVILLE DISTRICT	827,640
	MEMPHIS DISTRICT	93,157
	DENVER DISTRICT	538,484
	LOS ANGELES DISTRICT	1,212,312
	PORTLAND DISTRICT	181,777
	SAN FRANCISCO DISTRICT	105,209

Figure 6 - 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.

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

It is recommended that you start by copying the default Oracle10 style and modifying it rather than starting from scratch. The styles and skins are stored in individual folders located in the OracleBI\web\app\res folder on the Oracle Presentation Server machine. These files will also be duplicated in the web\app\res folder under your web application server as well.

Modifications to these files made in one location must be copied to the other location.

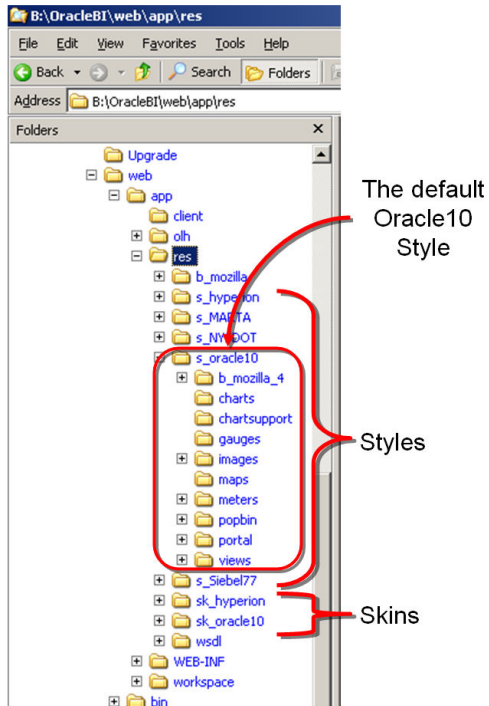


Figure 7 - Oracle Styles and Skins folder structure

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.

Working with the included Cascading Style Sheets

There are three files located in the b_mozilla_4 sub-folder in each style that affect most of the dashboard user interface:

PortalBanner.css Influences the overall appearance of a dashboard’s top section. This includes the dashboard’s names, links, and so on.

PortalContent.css Influences the overall appearance of a dashboard’s main content area.

Views.css Corresponds to each of the Oracle Business Intelligence request views (Title, Table, Pivot Table, Chart, Narrative, Ticker, and so on).

By modifying these files, developers can select default colors, contrast and fonts that may benefit users with certain disabilities.

Applying a Style to an Individual Dashboard

You might want to create a set of dashboards with content specifically optimized for users with accessibility needs. So you might also want to apply a special “accessibility” style to one or more individual dashboards for those users who need it.

To do this, select the target dashboard, click on the Page Options button on the right side of the screen and then select “Edit” mode. Once in the main dashboard edit screen, select the Dashboard Properties icon at the top left of the screen.

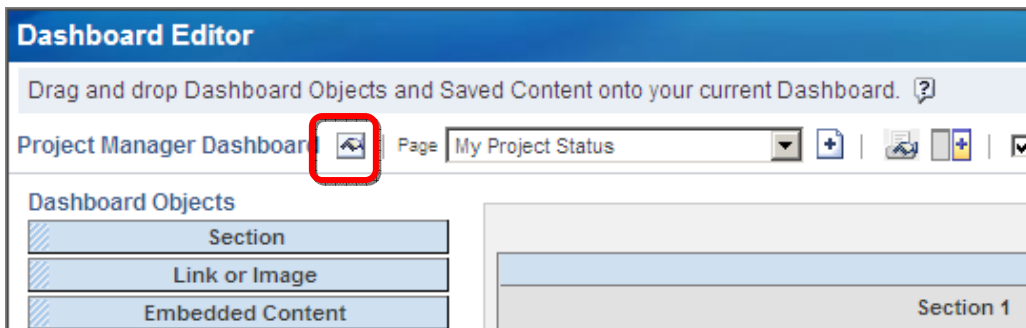


Figure 8 - Accessing the dashboard properties

The next screen you see 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.

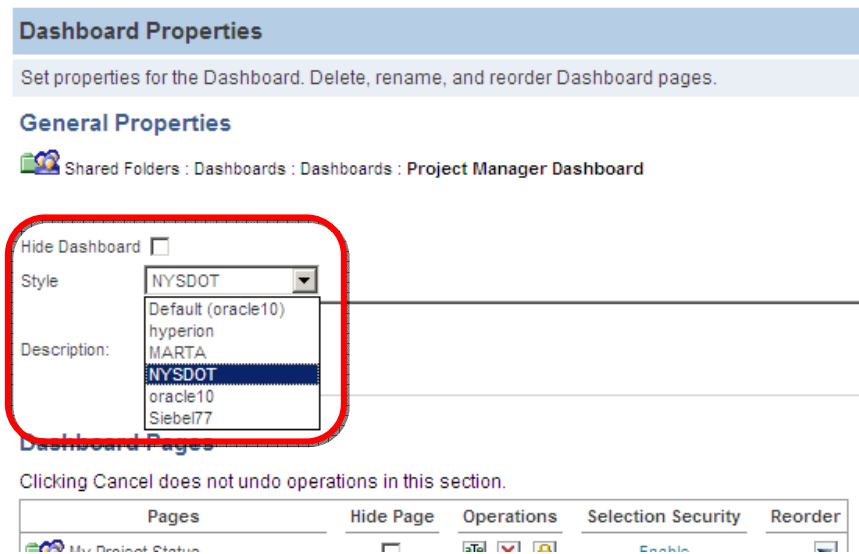


Figure 9 - Applying a Custom Style to a Dashboard

Use View Selectors to Switch Content

In some cases, you may want to create additional content specifically optimized for 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. You can use a View Selector and set it up as follows:

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.

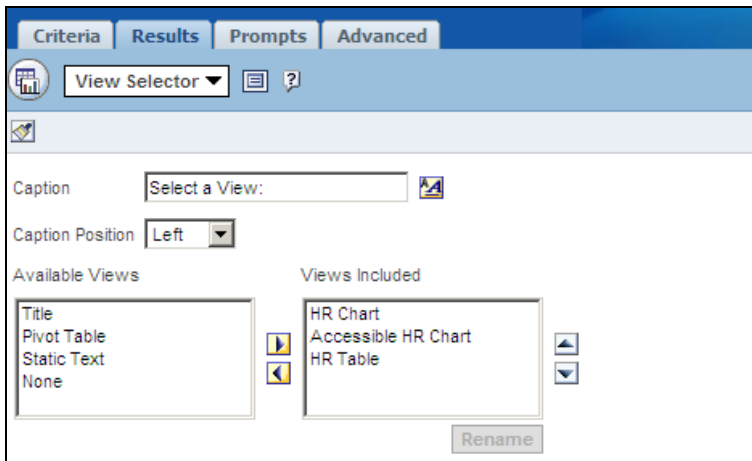


Figure 10 - View Selector Definition

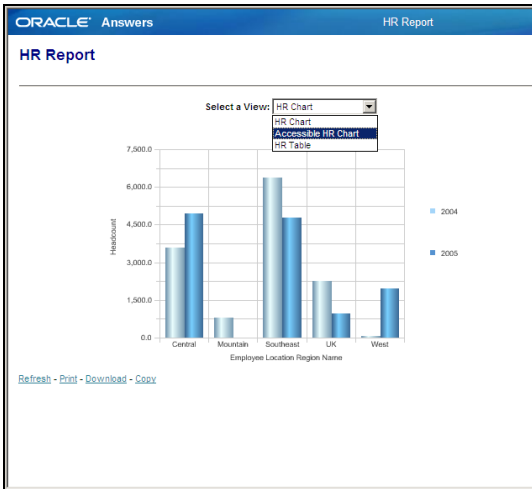


Figure 11 - View Selector in Action

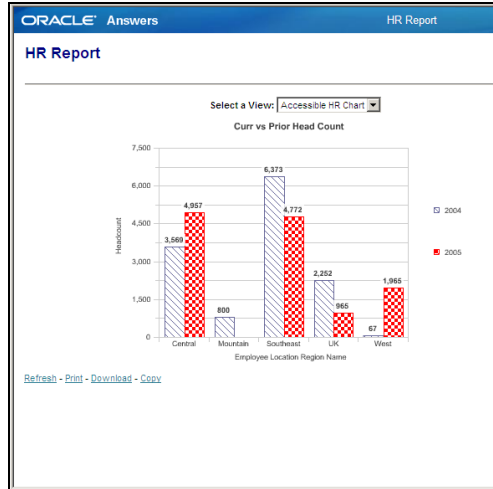


Figure 12 - After selecting accessible Chart version

In this 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. The accessible version also simplifies the background grid and adds a descriptive chart title.

Creating Dropdown Selections that have a GO Button

If you navigate within a dropdown using the up and down-arrow keys, the dropdown will automatically select the next value you arrow to. Many users with accessibility needs prefer that dropdown selection controls not execute any actions until a “Go” button has been selected. This makes navigation of dropdown lists easier to use with a keyboard.

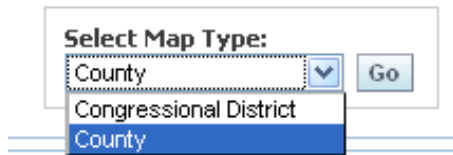


Figure 13 - Sample Dropdown with a GO Button

The only dropdown controls that have associated GO buttons are Dashboard Prompts. These prompts are typically used to select filter values for one or more reports displayed on a dashboard. You can alter this default behavior and use a Dashboard Prompt to provide display selection capability instead. This is accomplished by using a combination of multiple Dashboard

Sections, the Guided Navigation properties of those sections, an Answers Request that acts as a selection 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

You first design a Dashboard Prompt to create a Presentation Variable used to store which value the end user selects from the dropdown list. In this example, we are using a Presentation Variable named “p_map_type”, as shown below.

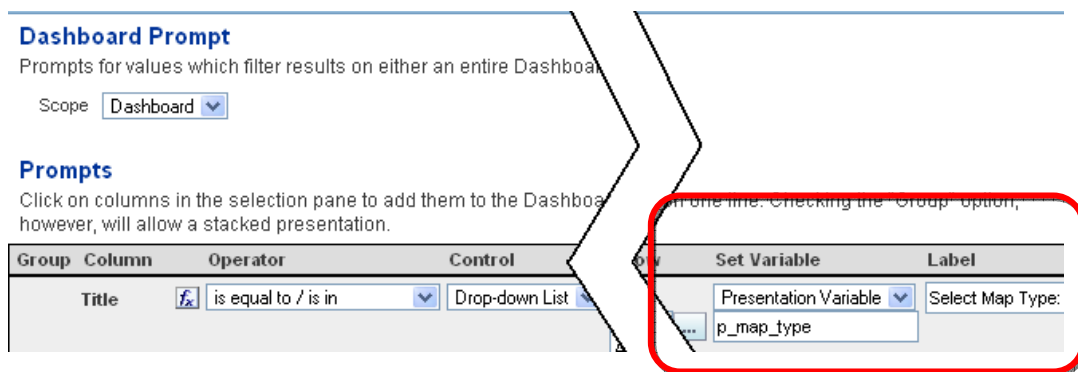


Figure 14 - Dashboard Prompt definition using a Presentation Variable

2) Create a “Visibility Controller” Request

Next, you create an Oracle Answers request that uses the value of the Presentation Variable as part of a filter to control whether the query request returns rows. Below is an example controller request that will return one or more rows if the user selects a value of “County” from the dropdown, and will return no rows if they select “Congressional District”.

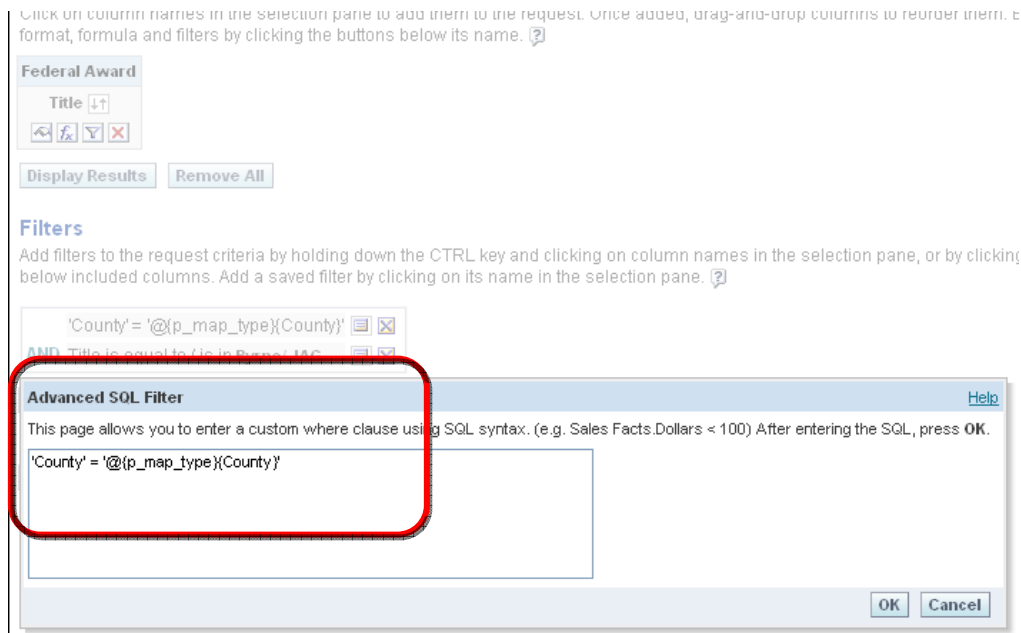


Figure 15 - Answers Request as Visibility Controller

In this case, an Advanced SQL Filter is defined in a way that references the value of the `p_map_type` variable so as to produce an evaluation of true when the value of the variable equals “County”. The filter is defined as follows:

‘County’ = ‘@{p_map_type}{County}’

It is recommended that you save all your controller requests in a separate sub-folder under the “shared” folder. This way, they won’t get confused with other query requests that are used to create data displays. In this example, we saved this controller request as “County Control” in the shared/Recovery Dashboard/Utilities folder.

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.

Your layout should look something like this:

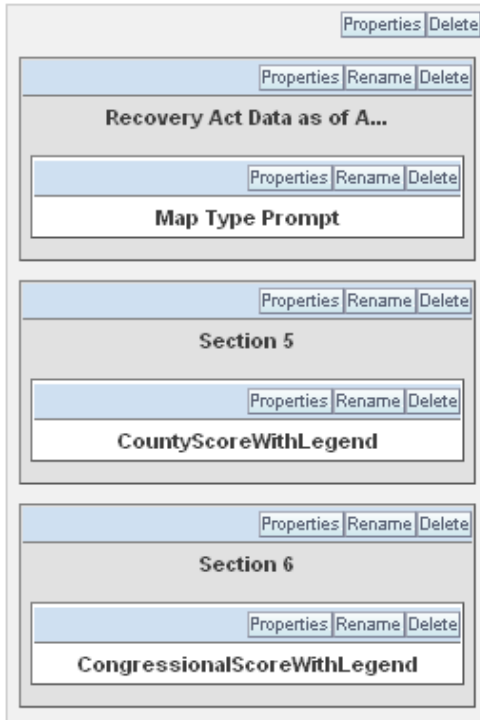


Figure 16 - Dashboard Section Layout Organization

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 Guided Navigation 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 section which you want to be visible when the user selects “County”, you click on the “Properties” button and then select “Guided Navigation”. On the subsequent screen, define the Source Request to be the controller Answers request named “County Control” and set the Guided Navigation “Show Section” property to “if request returns rows”, as shown below.

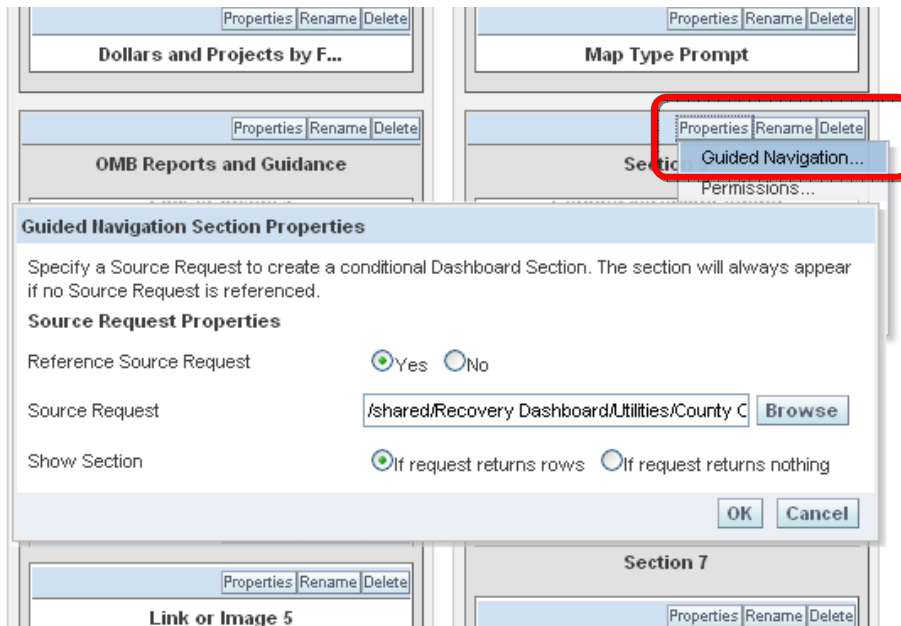


Figure 17 - Making the section's display conditional

For the display section that you want displayed when the user selects “Congressional District”, repeat the process but set the Guided Navigation “Show Section” property to be “if request returns nothing”.

Use Static Text objects instead of Title objects for 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 can not add other tags to the design of a Title object.

You can use the Static Text object to replace the Title object and supply the 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.

The screens in Figure 11 actually use this technique to create a report header that tags the Title as H3 text instead of as Title text, with a horizontal rule line underneath. This was defined using a Static Text object that had its text defined as:

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

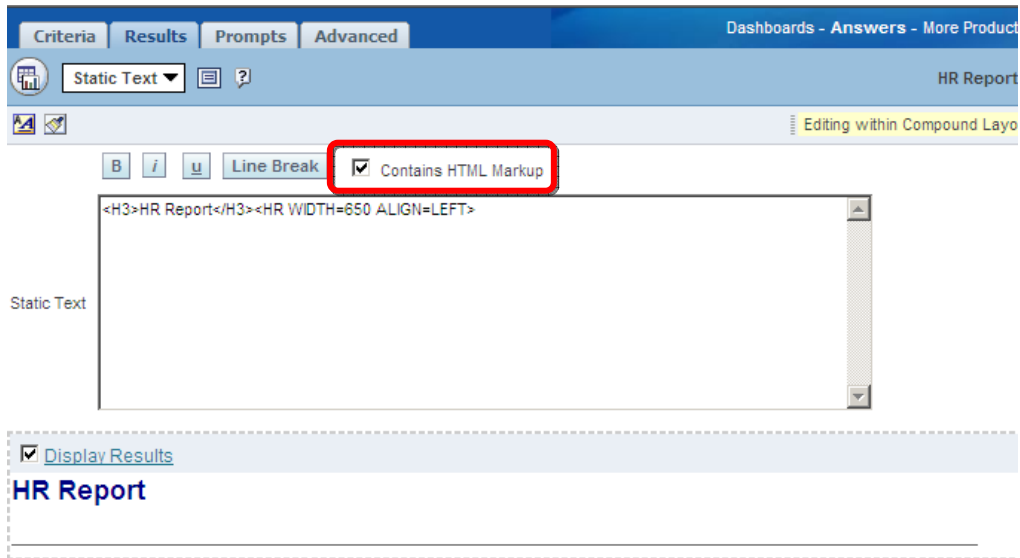


Figure 18 - Using HTML Mark-Up in a Static Text object

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.

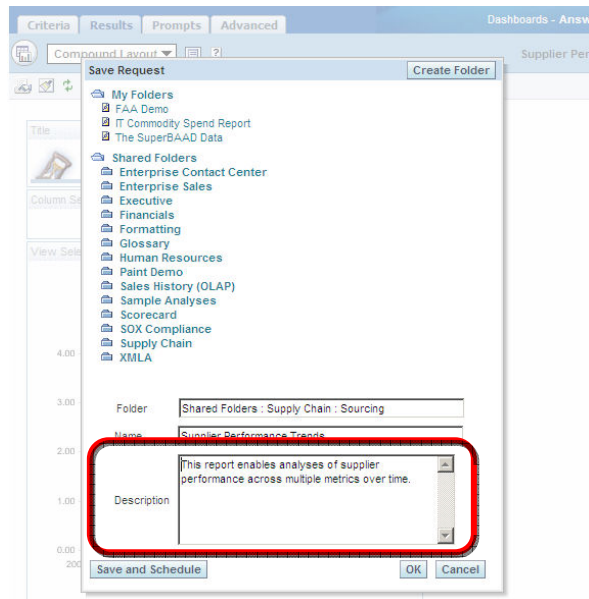


Figure 19 - Defining a Report's Description

BI Publisher: Turning on Accessibility Mode for Output

In addition to turning on the user's Accessibility Mode setting, the BI Publisher Runtime Configuration must also be set up to generate report output that is accessible to users with disabilities.

To turn this setting on, log in to BI Publisher as a user that has XMLP_ADMIN privileges, and then select the Admin tab. You will now be presented with the general BI Publisher administration screen. In the right-hand column, select the "Properties" link in the Runtime Configuration area.

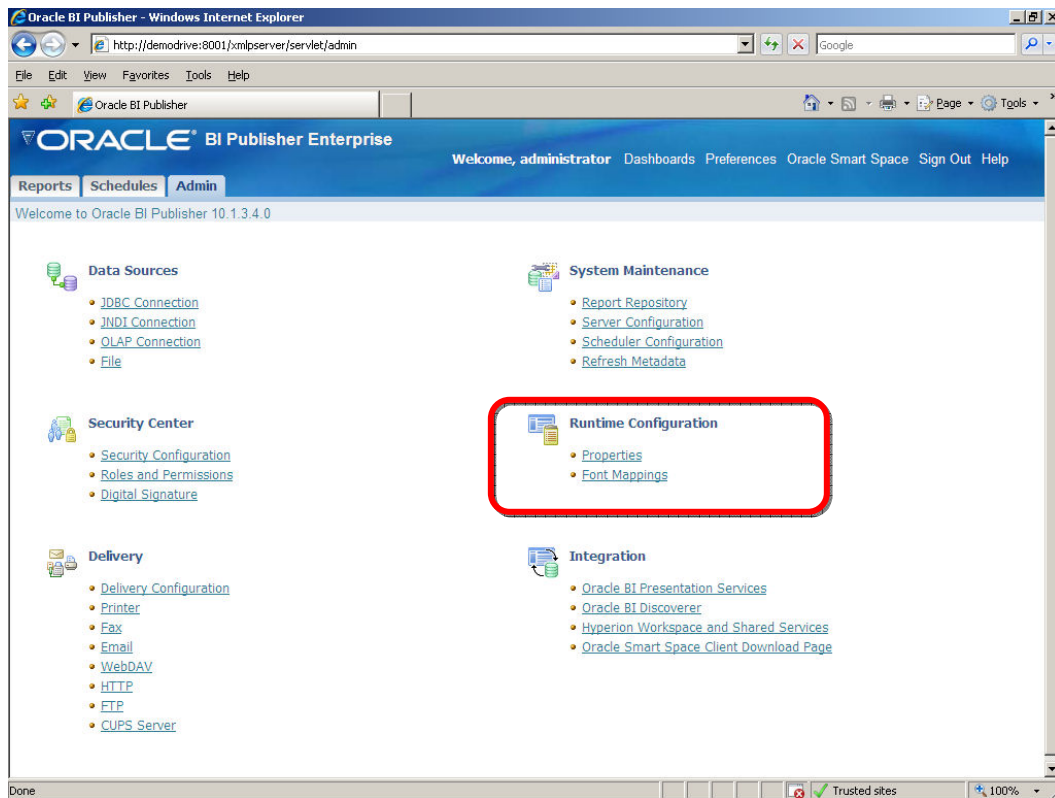


Figure 20 - The BI Publisher Admin screen

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.

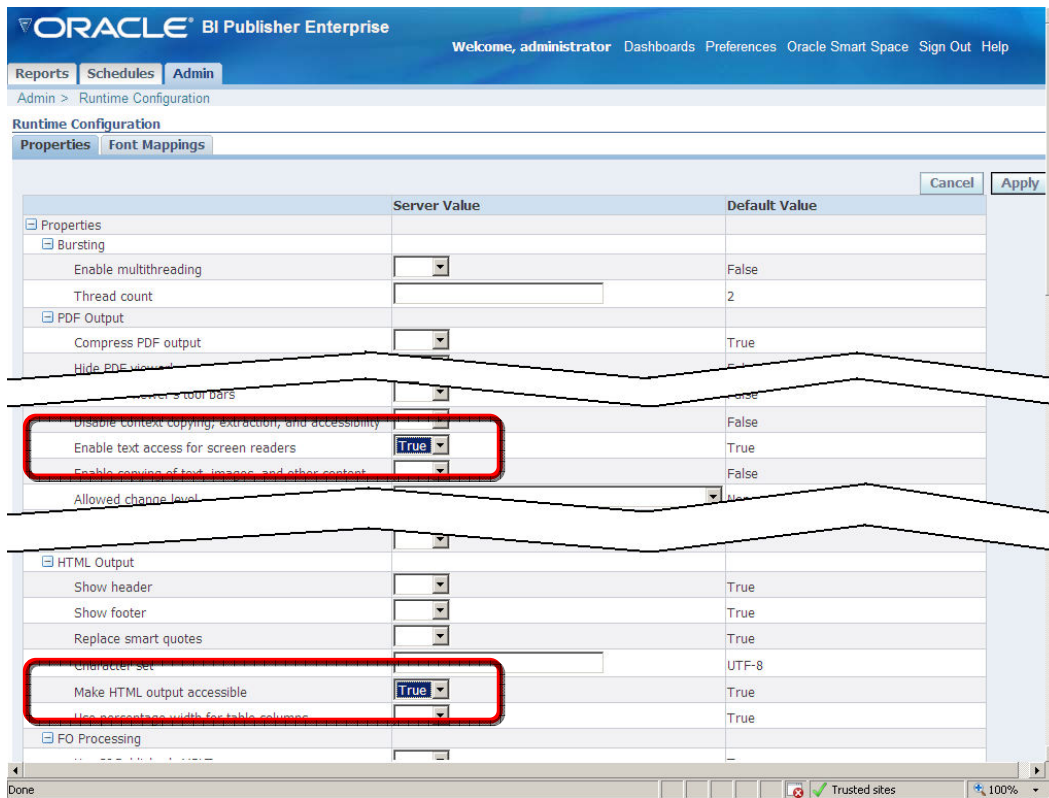


Figure 21 - Turning Accessibility Mode to True for PDF and HTML report output

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 (menu: File | 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 22 - 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. (Right-Click menu, then select “Properties”).
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.
- 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. To access this property in MS Word 2003, double-click on the image, or right-click then select “Format Picture”, and then select the “Web” tab. To access this property in MS Word 2007, right-click on the image, select “Size” and then select the “Alt Text” tab.

If you are 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 PDF output located at http://blogs.adobe.com/accessibility/assets/WordToPDFReferenceCard_v1.pdf.

IR & EPM Workspace: Turning on Accessibility Mode

Like the Oracle Dashboards module, the accessibility mode must be turned on for the EPM Workspace and Interactive Reporting. This mode in the EPM System, however, is set on an individual user basis. Once turned on, this mode will be automatically applied to all web-based interactions with both the EPM Workspace and the Interactive Reporting modules for users that have this property set.

Administrators can select two different settings for users with accessibility needs. Each user can be configured differently for each setting and the settings can be applied independently of each other. The specific feature settings are:

- Enable Screen Reader Support
This setting will auto-generate additional HTML markup to support specific screen reader features.
- Select Theme
This setting allows you to choose between a regular display theme and a high-contrast display theme.

EPM System Administrators can use the Shared Services Console to configure these user settings ahead of time. You access the Shared Services Console by logging into the EPM Workspace as a user with global administrator privileges. When you click on the navigation wheel in the upper left corner of the screen, you select “Administer”, then “Shared Services Console” from the menu that appears.

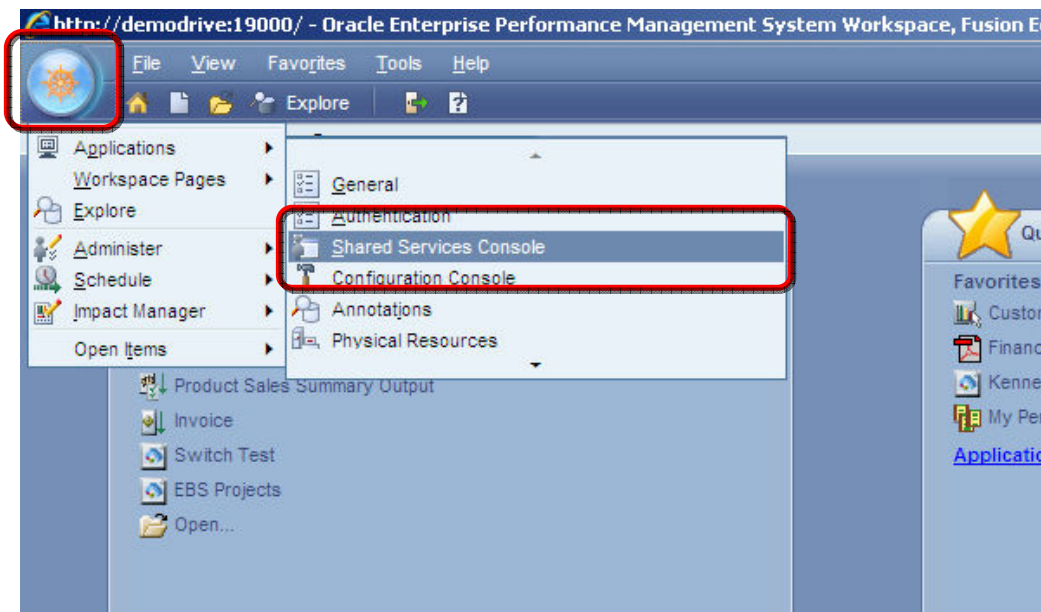


Figure 23 - Accessing the Shared Services Console

Once inside the Shared Services Console, open the Application Groups folder, and then open the Hyperion System 9 BI+ folder. Right-click on the specific server instance your users with accessibility needs will be using and select “Assign Preferences”.

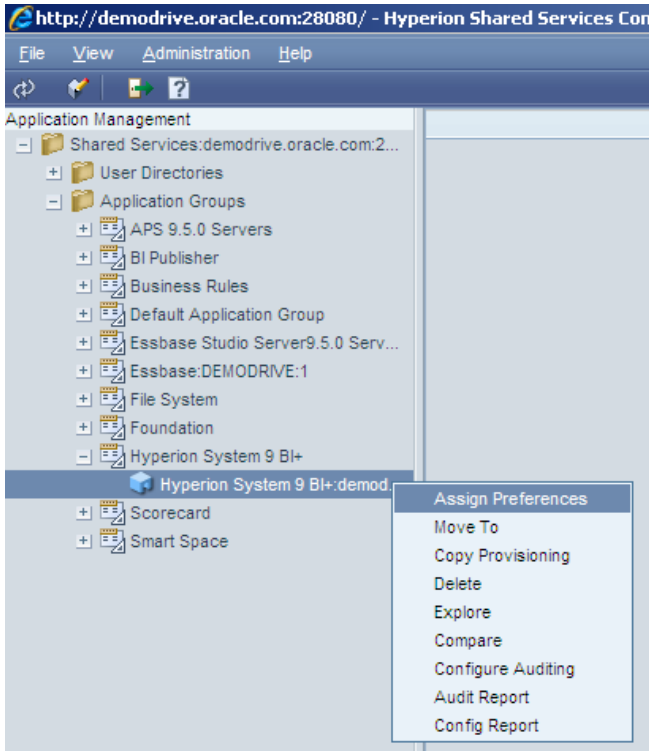


Figure 24 - Select the BI+ Application Group to set global preferences

In the screen which follows, make sure the display is set to Available Users, then add one or more users to the Selected Users and Groups list. Click on or select the “Continue” button to proceed to the next screen.

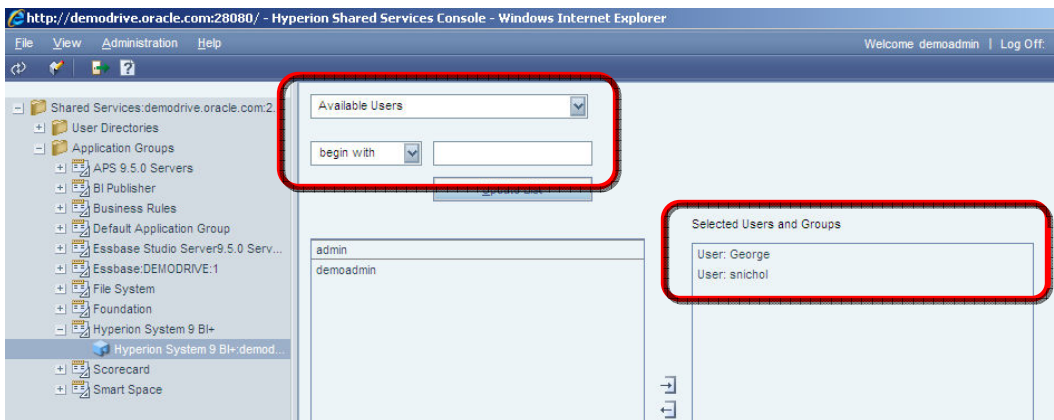


Figure 25 - Selecting the users with accessibility needs

In the next screen, select default folder locations and start page settings. Users with accessibility requirements must start in Explore mode, so make sure you select the “Explore” start option as well as a starting folder. These are not the default settings and must be selected explicitly. Finally, either turn on the Enable Screen Reader Support checkbox as shown, or select a theme. Click on or select “Continue” to save your selections.

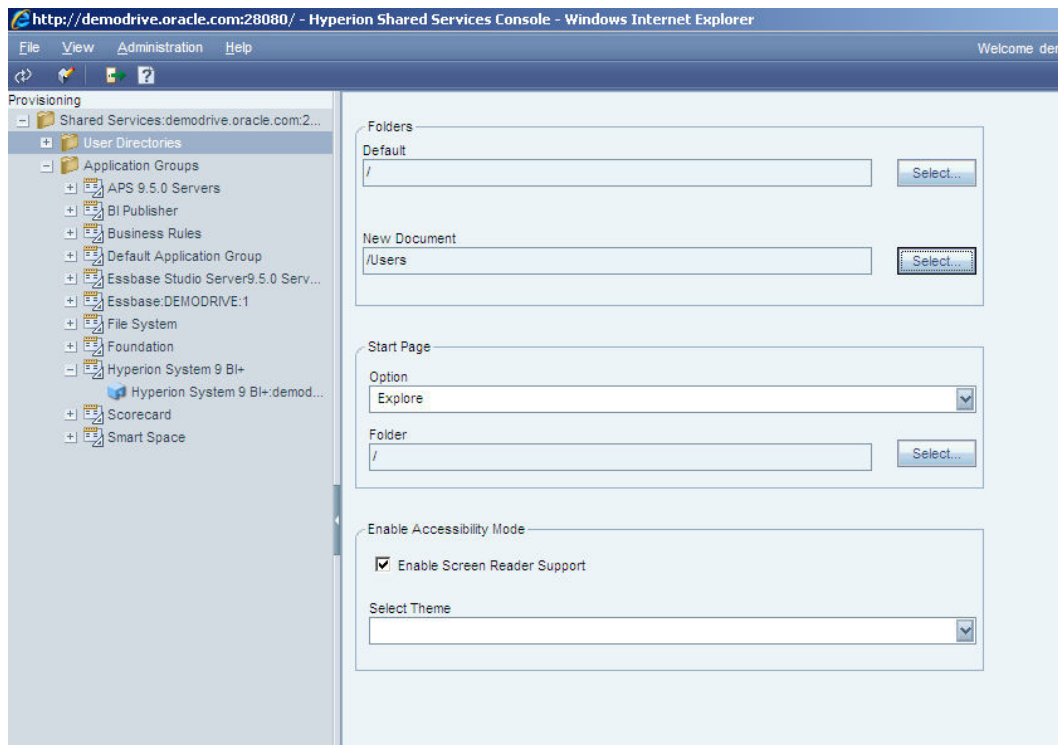


Figure 26 - Turning on Accessibility Mode for the user

IMPORTANT: The users you selected on the previous screen will all be assigned the same accessibility settings. If you have users that need different settings, select them as a separate group and then apply the settings required by that new group of users.

Remember that for new users, the default settings will have all EPM System accessibility settings turned off. Any new users that are added to the system later will need to be configured for accessibility either by a Shared Services administrator or by the user themselves.

Please go to <http://www.oracle.com/technology/documentation/epm.html> and reference the EPM Workspace Administrator’s Guide and Hyperion Shared Services Administration Guide for details on configuring and provisioning users and groups in the EPM Workspace.

Accessible Ad-Hoc Queries: Setting up for Casual Users

For most casual BI users, their definition of an ad-hoc query capability means they can select one or more fields to see, which types of data views to use and enter selection values for one or more common filters.

Oracle Dashboards can provide this capability for casual users without requiring them to drop into Oracle Answers or Interactive Reporting. You accomplish this by designing one or more “ad-hoc” dashboards that use a combination of column selectors, view selectors and dashboard prompts to create this kind of “casual ad-hoc” environment.

Accessible Ad-Hoc Queries: Setting up for Power Users

Power users want much more flexibility and control over the types of ad-hoc queries they can create. They also want the ability to create their own calculated columns and to leverage other advanced query features. The Interactive Reporting module is a power user query tool which supports accessibility and is included in the OBIEE+ license.

Interactive Reporting can use the same metadata for ad-hoc queries that Oracle Answers and Oracle Dashboards are using. To set this up, use the following steps:

1. Install the Oracle BI Server Client and the Hyperion IR Studio on a local developer’s desktop PC.
2. Use the IR Studio to create an OCE connection file that accesses the Oracle BI Server via the ODBC driver installed with the Oracle BI Server Client.

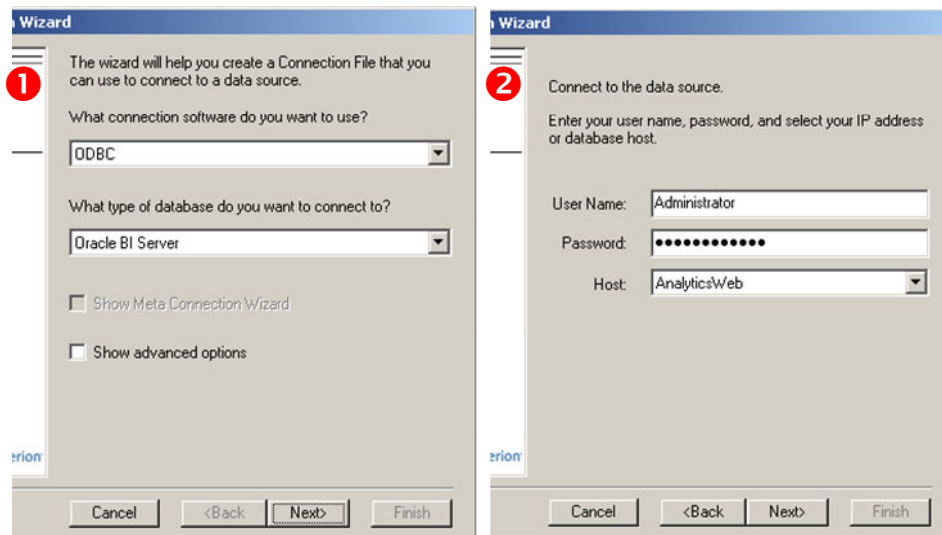


Figure 27 - Creating an Interactive Reporting Connection to the Oracle BI Server

- Use IR Studio to create a BQY reporting document with one or more general-purpose data models that use the appropriate OBIEE Subject Area(s). Make sure you turn off the auto-join option in the DataModel | Data Model Options... menu selection. IR queries against the Oracle BI Server must use un-joined data models, as shown below.

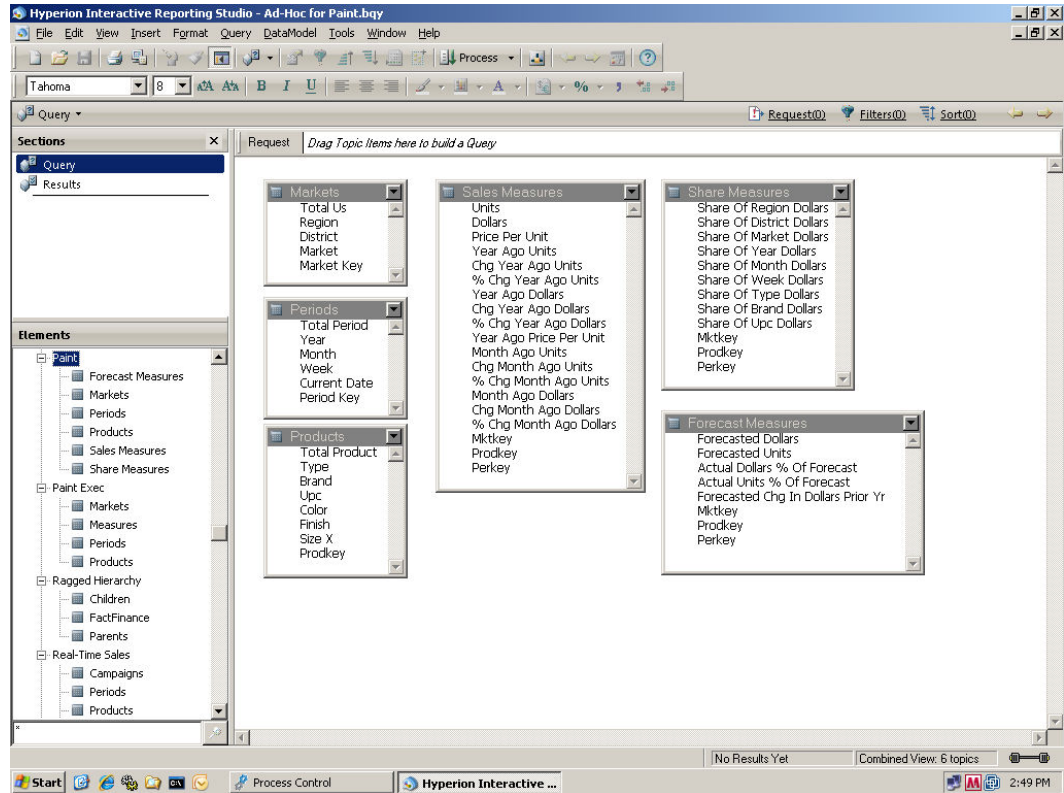


Figure 28 - Sample IR Template for Ad-Hoc Query support of the Paint subject area

The Interactive Reporting file format supports the ability to create multiple query sections. You may choose to add a query section for each OBIEE Subject Area defined, but Oracle recommends that you create a separate ad-hoc query template file for each OBIEE Subject Area. That way you can apply EPM Workspace security to the query templates to ensure only power users with accessibility needs have access to them.

- Install the Oracle BI Server Client on the server box where the Hyperion System 11 Data Access Service (DAS) has been installed.
- Configure the DAS to access the ODBC source created by the installation of the Oracle BI Server Client on the server machine. Make sure you use the exact same data source name, including case, as was set up during the Oracle BI Server client installation. (The Data Access Service parameter definitions are case-sensitive.)

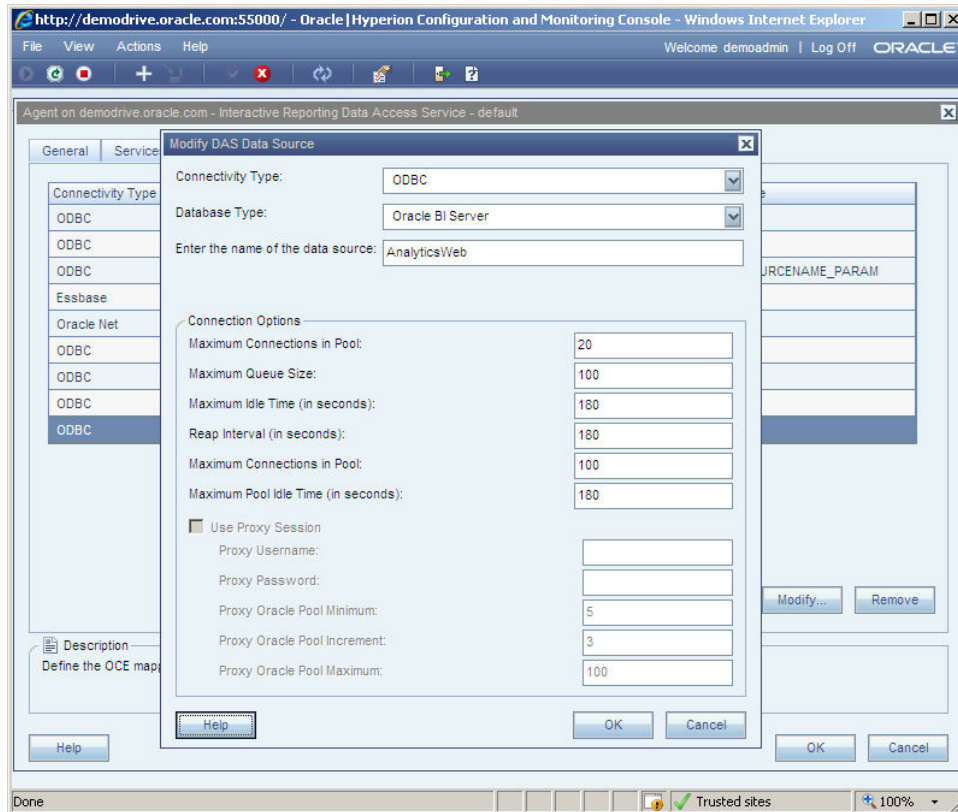


Figure 29 - Configuring the Data Access Service to use the Oracle BI Server

6. Log into the EPM Workspace as an admin user, and import both the OCE connection file and the BQY query document. Configure the BQY file to use the OCE connection file just imported.

In the Workspace-based screen shot of the query template in Figure 30 below, notice how the “tables” listed exactly match the logical tables that were included in the data model design of the Interactive Reporting query template shown in Figure 28. The tables and columns displayed under each logical table are the same as what an Oracle Answers user would see.

By creating this IR-based access to the Oracle BI Server metadata, your accessibility-needs power users now have the ability to create their own queries, tables, charts and pivots in a mode that supports full keyboard-only access as well as providing the additional HTML markup required to support screen readers and other accessibility technologies.

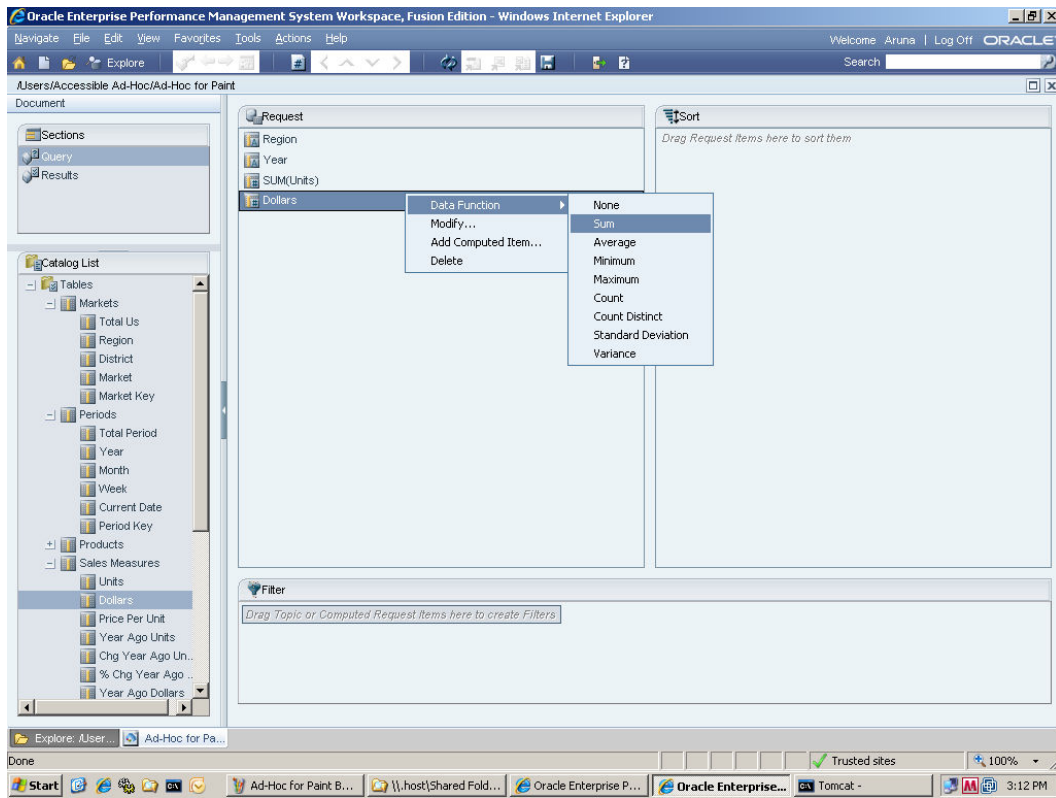


Figure 30 - Building a query in the Accessibility Mode of the EPM Workspace

For more complete details on how to create OCE connection files, BQY query documents, OBIEE-based data models within those documents, and then how to publish & configure the files in the EPM Workspace, please refer to the EPM Workspace Developer's Guide and the Hyperion Interactive Reporting Studio User's Guide documents. These can be found in the Foundation Services tab and Reporting and Analysis tab in the documentation set located on OTN at http://download.oracle.com/docs/cd/E12825_01/index.htm.

Accessible Scheduling: Setting up the EPM Workspace

Scheduling in the EPM Workspace is already accessible when the EPM Workspace user preferences are set as described above. To schedule actions that are designed to execute queries, simply import an Interactive Reporting BQY query document as a Job instead of as a File.

To schedule other OBIEE actions within the EPM Workspace, you will need to configure a generic job to execute an OBIEE API program. Please refer to the EPM Workspace Developer's Guide and the OBIEE Web Services Guide for further details.

Conclusion

The Oracle Business Intelligence Suite Enterprise Edition Plus 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 these guidelines 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
- 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 Enterprise Edition
December 2009

Authors: Kendall Garey, Ken Kipnes,
Mark Ostroff

Contributors: Tim Dexter, Larry Johnson,
Bryan Wise

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 © 2009, 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.