

# Accessibility in Oracle Forms Applications

ORACLE WHITE PAPER | AUGUST 2016





## Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®



## Table of Contents

Disclaimer	1
Introduction	3
Running a form if you are a user with disabilities	3
Configuration steps for screen reader users	4
Oracle E-Business Suite features	4
Runtime options for users with low vision	5
Look and Feel and Color Scheme	5
Read Only Fields	6
Font Size	6
Oracle E-Business Suite features	7
Configuring Forms to integrate with a screen magnifier	7
Configuring Forms to integrate with voice recognition	7
Runtime options for users with physical limits	7
Hints for Voice Activation	9
Oracle E-Business Suite features	9
Runtime Forms Parameters	10
Building forms for users with disabilities	10
Coding for screen reader users	11
Prompts	11
Display Items	11
Attributes Available for Each Item Type	11



Special-case Prompts	12
Matrix Layouts	13
Using Display Items to mimic Prompts	13
Graphics Text	14
Additional Attributes	14
Coding for users with physical limits	14
Access Keys	14
Accelerators	14
Exceptions for Not Having Access Keys	15
Oracle E-Business Suite features	16
Blocks	17
Tab Pages	17
Menus	17
Coding for users with low vision	17
The 'automatic' color	17
Oracle E-Business Suite features	18
Testing	18
Versions and known issues	18
Conclusion	18



## Introduction

Oracle's goal is to ensure that products and services are accessible to all users with good usability and adhere to Section 508. Industry standards will continue to evolve over time and Oracle is actively engaged with other market leading technology vendors in addressing apparent technical hurdles. Accessibility to the increasingly complex technology pervasive in our lives is a factor for everyone. Accessibility solutions for users with disabilities have often become mainstreamed productivity enhancements that benefit everyone.

This paper contains instructions for users with disabilities on how to configure their system and operate forms in the most efficient manner. Also included is information regarding how to use Oracle Forms 6i, 9i, 10g, 11g and 12c to build applications that are accessible to the broadest possible range of users and adhere to Section 508. Techniques are covered for Form development using the Oracle Forms product by itself, or in conjunction with Oracle E-Business Suite.

This paper addresses applications that are deployed on the web, as opposed to client/server applications. Several techniques discussed, especially those regarding colors, are only available in the web-deployed mode. Examples of specific keystroke combinations are for the Microsoft Windows operating systems; the sequences will vary slightly on other platforms.

As with any standards or design recommendations, following these rules does not guarantee that a product is accessible or usable. Ultimately there is no substitute for testing a product with the intended target audience.

## Running a form if you are a user with disabilities

This section assumes that the user is operating screens that are running on the web which were developed with Oracle Forms 6i, 9i, 10g, 11g or 12c according to all of the principles outlined in this white paper under heading 'Building forms for users with disabilities'.

Users with low vision can:

- » Run forms with any color scheme set on the operating system by using the Generic Look and Feel or use one of several pre-defined schemes with the Oracle Look and Feel
- » Turn off hard-coded colors
- » Set operating system Font Size with DPI which affects the overall size of all items in a form
- » Use a screen magnifier that supports Java such as ZoomText, MAGic or SuperNova

Users with physical limits can:

- » Run forms with just the keyboard
- » Use access keys to activate menu items, push buttons, radio buttons, checkboxes
- » Invoke the "List Tab Pages" function (typically mapped as F2) to switch tab pages

- » Change forms keystroke mappings that are displayed in the "Keyboard Help"
- » Use operating system accessibility features such as Sticky Keys and Toggle Keys
- » Use a voice recognition program such as Dragon Naturally Speaking to give commands and enter data

Screen reader users can:

- » Run forms with just the keyboard
- » Use access keys to activate menu items, push buttons, radio buttons, checkboxes
- » Invoke the "List Tab Pages" function (typically mapped as F2) to switch tab pages
- » Change forms keystroke mappings that are displayed in the "Keyboard Help"
- » Use operating system accessibility features such as Sticky Keys and Toggle Keys
- » Run forms with a screen readers that supports Java such as JAWS or SuperNova
- » Use Oracle E-Business Suite features that display all items and push buttons in a window
  - » Key-F9 function -- Prompt/Value List of Values (LOV) (typically mapped as Ctrl+Shift+F9)
  - » Key-F8 function -- Actions LOV (Typically mapped as Ctrl+Shift+F8)
- » Use Oracle E-Business Suite feature called Forms Personalization to change "speakable prompts".

Users with hearing impairments can:

- » Turn on the operating system accessibility feature SoundSentry to generate a visual warning when the system makes a sound

### Configuration steps for screen reader users

Oracle Forms supports the Java Access Bridge, which allows integration with screen reader assistive technologies that also support Java. By default, the Java Access Bridge is not enabled. For information how to enable the Java Access Bridge, please see the [Java Accessibility Guide](#). The Java Access Bridge must be enabled so that Oracle Forms, the Java Runtime Environment (JRE) and the Java enabled screen reader may interact.

See heading 'Coding for users with physical limits' in this paper for keyboard-only usage techniques.


### Oracle E-Business Suite features

#### *Actions and Values LOVs*

Oracle E-Business Suite incorporates a feature that allows any user to see the current screen in a compressed, text-only popup window format called LOV (List of Values). Fields which cannot take focus because they are 'non-navigable' will not allow a screen reader to read their value and prompt. To account for this, Oracle E-Business Suite has special code that presents all fields in the current window, as well non-navigable fields in the window in special LOVs. Included in the LOVs are the values of display items, which otherwise would not be easily discernible with a screen reader because they are not keyboard navigable. These special text-only popup windows allow a screen reader user to quickly identify all widgets in the current window (but just the current row for multi-row blocks).

The 'Actions' LOV is invoked through the "KEY-F8" function and is a list of all push buttons in the current window. The 'Values' LOV is invoked through the "KEY-F9" function and is a list of all other widgets in the current window like text items, radio buttons, checkboxes and poplists. Each row in the LOV will be spoken by a screen reader. The LOVs are in alphabetical order. Both LOVs also show access keys for radio buttons, checkboxes and push buttons. Choosing a value from either the Actions or Values LOV will not cause focus to move to those fields or buttons.

The access keys displayed in the LOVs are within braces for translation purposes. For example, access key c is displayed as {C} and a screen reader will speak the text as "brace C brace". Check with the screen reader manufacturer if there is a way to change it to speak "Alt C" instead of "brace C brace" if this is annoying.



Note that the “KEY-Fn” function is not necessarily the “Fn” button on the keyboard. The current key mapping for the function can be shown in the “Keyboard Help” window. Typically the “KEY-Fn” function is mapped to Ctrl+Shift+Fn via the Oracle Terminal resource file.

The Oracle Forms generic code for Actions and Values LOV that can be used by non-Oracle E-Business Suite developers to code similar functionality is available upon request.

#### *Forms Personalization*

Oracle E-Business Suite users can take advantage of a powerful feature called Forms Personalization if they don't like some of the “speakable prompts”. The Form Personalization feature allows you to declaratively alter the behavior of Forms-based screens, including changing properties, executing builtins, displaying messages, and adding menu entries. My Oracle Support Note 395117.1 is available that explains Forms Personalization.

#### Runtime options for users with low vision

##### **Look and Feel and Color Scheme**

Oracle Forms can be run with either the Oracle Look and Feel or the Generic Look and Feel. The Oracle Look and Feel consists of a new look and feel for each item, and a pre-defined set of color schemes. The Generic Look and Feel adheres to the native interface and color scheme of the current operating system. The choice of look and feel has an impact on accessibility features; in general, the Oracle Look and Feel is the more accessible of the modes. Only users with low vision should find it necessary to use the Generic Look and Feel to control the overall colors of the application, either to increase or decrease contrast.

Users with low vision may set the desired colors using the operating system's provided schemes, then specify lookAndFeel=generic in the URL so that Oracle Forms will use these colors.

- » To specify the look and feel, set the following parameter when launching a form: lookAndFeel= either 'generic' or 'oracle'
- » If the Oracle Look and Feel is used, the color scheme can be specified as follows:
  - » colorScheme= one of teal, titanium, red, khaki, blue, olive, or purple
  - » The colorScheme parameter has no effect if lookAndFeel is set to 'generic'.

An example of a form run with lookAndFeel=oracle and colorScheme=blue:

Responsibility	Application	Security Group	From	To
Application Developer	Application Object L	Standard	29-JUL-1999	
Workflow Administrator	Application Object L	Standard	10-NOV-2000	
Workflow Administrator W	Application Object L	Standard	10-NOV-2000	
System Administrator	System Administrati	Standard	01-JAN-1951	
General Ledger Super Use	Oracle General Led	Standard	17-SEP-1999	

Figure 1: The 'Users' form from Oracle E-Business Suite running in the Oracle Look and Feel. The primary canvas color is blue, with tab controls drawn in gray. All widgets have a rounded appearance.

The same form run with lookAndFeel=generic on the Microsoft Windows operating system:

Responsibility	Application	Security Group	From	To
Application Developer	Application Object L	Standard	29-JUL-1999	
Workflow Administrator	Application Object L	Standard	10-NOV-2000	
Workflow Administrator W	Application Object L	Standard	10-NOV-2000	
System Administrator	System Administrati	Standard	01-JAN-1951	
General Ledger Super Use	Oracle General Led	Standard	17-SEP-1999	

Figure 2: The 'Users' form from Oracle E-Business Suite running in the Generic Look and Feel. All canvasses are rendered in tan. All widgets have the standard Microsoft Windows appearance.


### Read Only Fields

Oracle Forms supports a feature where fields that cannot be entered (read-only fields) are automatically rendered dark gray. To turn this feature off, specify readOnlyBackground=false in the URL.

### Font Size

Operating system settings such as Font Size, will affect the overall size of all items in a form. Often this is the only technique to adjust font sizes within a form, as they typically are hard-coded.





Microsoft Windows 7 operating system instructions for changing the font size:

1. Go to Windows Control Panel > Fonts > Change font size > Set custom text size (DPI) = 200%
2. Restart the computer
3. Launch Forms and you should have a larger font.

### Oracle E-Business Suite features

*Profiles 'Java Look and Feel' and 'Java Color Scheme'*

With Oracle E-Business Suite, Oracle Forms screens may be invoked by actions taken on the Oracle E-Business Suite Professional User Interface or other self-service screens. In order to run these screens, a URL is constructed using information in profiles 'Java Look and Feel' and 'Java Color Scheme'.

- » To specify the look and feel, set profile 'Java Look and Feel' to either generic or oracle.
- » If the Oracle Look and Feel is used, the profile 'Java Color Scheme' can be specified as follows: teal, titanium, red, khaki, blue, olive, or purple.
- » The 'Java Color Scheme' profile has no effect if 'Java Look and Feel' is set to generic.

*Profile 'FND: Indicator Colors'*

Oracle E-Business Suite by default renders:

- » required fields in yellow
- » queryable fields in a different color while in enter-query mode
- » fields that cannot be entered (read-only fields) in gray

To turn off these features when running Oracle Forms through the Oracle E-Business Suite Professional User Interface, set profile 'FND: Indicator Colors' to No.

### Configuring Forms to integrate with a screen magnifier

Oracle Forms supports the Java Access Bridge, which allows integration with screen magnifier technologies that also support Java, such as ZoomText from AiSquared. Please refer to <http://www.aisquared.com/support/> for more information.

### Configuring Forms to integrate with voice recognition

Dragon Naturally Speaking supports Java and nothing special needs to be done to configure the software.

### Runtime options for users with physical limits

All items used within Oracle Forms follow the standard operating system conventions for keyboard use. For example, on the Microsoft Windows operating systems use Alt+<letter> to activate items with access keys, Alt+down to open a poplist, and Alt to move focus to the menu. Oracle Forms should inherit operating system accessibility functions such as Sticky Keys. Tabs can be switched by invoking the "List Tab Pages" function (typically F2), in addition to using access keys on each tab label.

The "Keyboard Help" window displays the keystrokes to achieve normal Forms operations, such as 'Next Block' and 'Clear Record'. This window can be viewed at any time by pressing Ctrl+K. The keyboard mappings can be customized as follows:

- » The System Administrator must locate the Oracle Forms resource file on the middle tier, typically called fmrweb.res
- » Make a copy of the file, naming it whatever you want, and locate it in the same directory as the original.

- » Open the new file in any text editor and make the desired keystroke mapping changes. Comments at the top of the file clearly explain how the mappings are performed.
- » To run this new mapping file, include term=file in the URL, where file specifies the complete path in addition to the filename.

A user running a screen reader will most likely need a modified keyboard mapping file, or will have to change the Assistive Technology keystrokes, as some of the default function mappings may conflict.

**TABLE 1: COMMON DEFAULT FORMS KEYSTROKES ON MICROSOFT WINDOWS**

Action	Keystroke
List of Forms Keys	Ctrl+K
Next Field	Tab
Previous Field	Shift+Tab
Next Block	Shift+PageDown
Previous Block	Shift+PageUp
Actions LOV	Ctrl+Shift+F8
Values LOV	Ctrl+Shift+F9
Activate default push button in a window if one exists	Enter Pressing the Enter key with the focus on a button will activate that button. If the focus is not on a button (or menu item), then Enter should activate the default button if one exists.
Save Record (Commit)	Ctrl+S
Clear Record	F6
Create Record	Standard keystroke may be consumed by screen reader, need to run with different terminal resource file to map Ctrl+Down to something else or just use the pulldown menu.
Close Window	Ctrl+F4
List of Tab Pages	F2
Activate Menu	Alt and then navigate with up/down and left/right arrow keys
Activate push buttons, radio buttons, checkboxes and topmost menu items	Alt+access key
Toggle between open/close poplist	Alt+Up/Down arrow keys
Activate current push button, Toggle checkbox yes/no	Spacebar
Cycle through and select a radio button within a radio group	Left/Right arrow keys and then Spacebar
Move to beginning of line	Home
Move to end of line	End
Select to end of line (there is no keystroke for	Shift+End / Shift+Insert+End

"Select All")

Cut	Ctrl+x
Copy	Ctrl+c
Paste	Ctrl+v
Select the current tree node	Spacebar
Activate the current tree node (this expands or contracts the node if it is not a leaf node)	Enter
Move up the tree	Up arrow key
Move down the tree	Down arrow key
Move up the tree branch one parent at a time collapsing nodes	Left arrow key
Move down the tree expanding parent nodes	Right arrow key
Move focus within tree without selection	Ctrl+Up/Down arrow keys
Move focus up/down the tree branch but not the selection	Ctrl+Left/Right arrow keys

### Hints for Voice Activation

You may be using the Dragon Naturally Speaking (DNS) "Tab Key" command and find yourself 'stuck' in a region of a form. If this happens, try using DNS mouse commands e.g. "mouse grid" or "move mouse + (direction)", to move to another region in the form. If the form has frames, you can use the DNS commands "Next Frame" or "Previous Frame".

Sometimes using the DNS "Press Key" command doesn't work. e.g. "Press Alt + C" or "Press Ctrl + O". If this happens try using the DNS mouse commands to move the cursor over the button or drop down list and then using the "Mouse Click" or "Press Enter" command to activate the button. The DNS command "Press Ctrl L" almost always works for drop down boxes (often designated by a box containing ellipsis).

There is no alt-text on the buttons. This is not HTML. They are labels. If there is an access key for the button, you can use command "Press Alt <letter>". If there is no access key then you can say "Tab" multiple times until focus is on the button and then use command "Press Enter".

If DNS stops taking voice commands, try using DNS mouse commands e.g. "Move Up" and "Move Down" and also "Press Enter" to expand a tree branch. Command "tab" works too.

The DNS mouse commands "Move Mouse Up, Down, Left, Right, Drag Mouse Up, etc." are useful. The scroll bar can be controlled in a form using these commands.

### Oracle E-Business Suite features

*Profile 'Forms Keyboard Mapping File'*

Oracle E-Business Suite provides profile 'Forms Keyboard Mapping File'. To run a new mapping file, specify the complete path in addition to the filename for the profile. When running Oracle Forms through the Oracle E-Business Suite Professional User Interface, the new mapping file will be used.

### *List of Values*

Oracle E-Business Suite includes a feature that renders an iconic button next to each field that has an LOV. The LOV can also be invoked from the keyboard by pressing the “List of Values” function (typically Ctrl+L).

### *Tab Pages*

Tabs in Oracle E-Business Suite can only be changed from the keyboard using the “List Tab Pages” function. Individual tab labels do not have access keys due to translation issues.

### **Runtime Forms Parameters**

There are many runtime parameters that can be set. They are documented in the Forms Deployment Guide at <http://www.oracle.com/technetwork/developer-tools/forms/documentation>.

## **Building forms for users with disabilities**

The Oracle Forms 6i Builder is not accessible. Oracle Forms 9i, 10g, 11g and 12c builders are accessible. The Oracle Forms 6i, 9i, 10g, 11g and 12c runtime are accessible if coded to the standards in this paper. By following the guidelines in this paper, developers can build runtime products that adhere to Section 508.

In order to meet the needs of all users, all forms should have the following capabilities:

- » A screen reader should have access to the value and ‘prompt’ for each item, so that a screen reader user or user with low vision can hear it. In this paper that is referred to as the “speakable prompt”.
- » All functionality should be accessible from the keyboard only, so that a user does not have to manipulate a mouse or other pointer device.
- » All color usage should be under the control of the user, so that a user with low vision or color blindness can see all of the content.
- » There should be no reliance on timed functions, so that a user does not need to respond in a set amount of time.
- » Flashing or animations should be avoided, or if present allow a mode that disables them.

Oracle Forms provides sufficient attributes to meet these accessibility requirements, namely:

- » When an item takes focus, Oracle Forms passes the current value, the speakable prompt and other attributes of the item to a screen reader.
- » Every item is operable from the keyboard, either by allowing navigation to it or allowing invocation via access keys.
- » Every item can specify ‘automatic’ as the color at design time, so that at runtime, the right color is chosen from the user’s control panel settings.

For issues of timed responses and animations, this paper assumes that they will simply be excluded from any screen design.

Using the bell built-in as an only indicator of an error is allowed. The operating system typically has some kind of accessibility option for sound such as SoundSentry that generates visual warnings when the system makes a sound.

For any BeanAreas that are coded within a form, the general principles outlined above need to be extended as appropriate to the Java code.

## Coding for screen reader users

- » Every item should have a Hint Text, Prompt, Label or Tooltip Text in order for a screen reader to have a speakable prompt.
- » Uniqueness of speakable prompts is important to the screen reader user since other clues such as physical location on the screen are lost.

A screen reader user also needs all of the features listed under the 'Coding for users with physical limits' heading of this paper.

### Prompts

Oracle Forms identifies speakable prompts in the following order of precedence:

1. Hint Text
2. Prompt
3. Label
4. Tooltip Text

Starting with Hint Text, the first attribute that is not null (after stripping any leading and trailing spaces) is interpreted as the speakable prompt and sent to the screen reader. Not every one of these attributes is available on each item, in which case that attribute is simply skipped.

Note that the property "Display Hint Automatically" should be set to 'No' if relying on Hint Text, otherwise any Hint Text added will appear on the message line when any user operates the product, and it will be spoken again by the screen reader.

### Display Items

Display items are not keyboard navigable but Oracle E-Business Suite uses them because the prompts and values can be displayed in a popup window in a text-only format.

### Attributes Available for Each Item Type

The following table lists the attributes available for each item (Available), the preferred attribute(s) to use for each item (Preferred), and non-applicable attributes (N/A):

**TABLE 2: ATTRIBUTES OF ITEMS**

Item	Hint Text	Prompt	Label	Tooltip Text
Text item	Available	Preferred	N/A	Available
Checkbox	Available	Preferred	Preferred	Available
Poplist	Available	Preferred	N/A	Available
Display item	N/A	Preferred	N/A	Available
Radio Group	Preferred	N/A	N/A	Available
Radio Button	N/A	Preferred	Preferred	N/A
Button (textual)	Available	Available	Preferred	Available



Button (iconic)	Available	Available	Available	Preferred
Tree	Available	Preferred	N/A	Available

**Special-case Prompts**

*'Obvious' Prompts*

There are some fields that have no displayed Prompt because the field's purpose is "obvious", typically because of its physical location on the screen.

Example:

The 'high' field of a low/high range, such as:



Figure 3: a portion of a screen showing the prompt 'Ship Dates', followed by a text item, a dash, then a second text item.

For this case, supply text such as 'Ship Dates: High' for the Hint Text, or a hidden Prompt, on the unlabeled field to serve as the speakable prompt.

*Non-unique Prompts*

Some fields have a prompt that is not unique or meaningful by itself, such as:

"Associative" object names:



Figure 4: a portion of a screen showing the prompt 'Vendor Name', followed by a text item, then the prompt 'Number' followed by another text item.

Instances in which the Title differentiates:

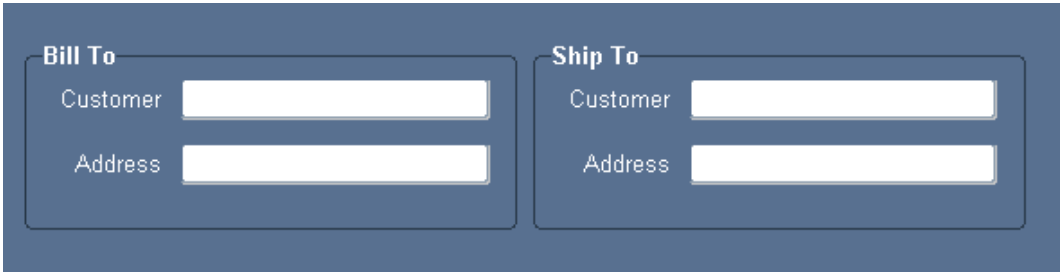


Figure 5: a portion of a screen showing two frames, one titled 'Bill To', the other 'Ship To'. Within each frame are two text items, the first with prompt 'Customer', the second with prompt 'Address'.

Instances in which the Field Names are not meaningful without Title:

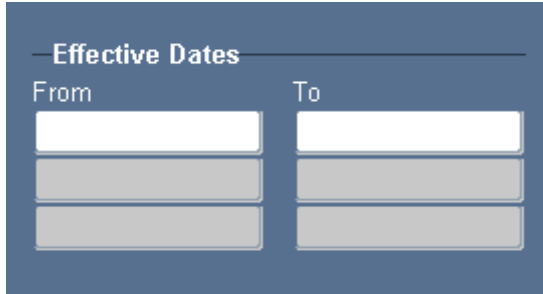


Figure 6: a portion of a screen showing the title 'Effective Dates' drawn in front of a horizontal line. The line spans two columns of a multi-row block with prompts 'From' and 'To'.

These fields should all use the Prompt attribute for the on-screen prompt, and also have Hint Text for the speakable prompt. For example, for the 'Customer' field in the 'Bill To' section, the Hint Text could be 'Bill To: Customer'.

### Matrix Layouts

Another common construct is 'matrix' style fields:

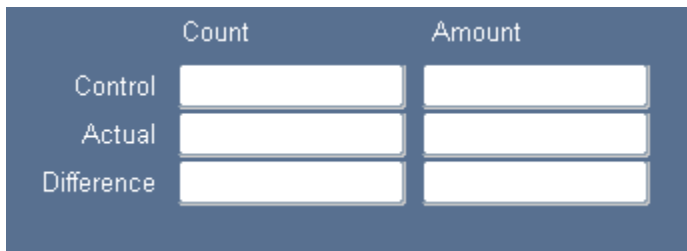


Figure 7: a portion of a screen showing 6 Text Items arranged in a grid format of two columns and three rows. The columns are labeled 'Count' and 'Amount', and the rows have labels 'Control', 'Actual' and 'Difference'.

In this case, Boilerplate (Graphics text) and Prompts should be used to label the fields on-screen. Each field should have Hint Text that combines the 'row' and 'column' prompts, for example, 'Control: Count'. If this is a multi-row block then the Hint Text needs to be changed dynamically in the WHEN-NEW-ITEM-INSTANCE trigger to reflect the prompt of the current record.

### Using Display Items to mimic Prompts

Avoid techniques that employ a display item to mimic dynamic prompts except when absolutely necessary, such as:

- » Prompts that need to be shown with special characteristics, such as a bevel (the real Prompt attribute has no such property).
- » Desirable 'clipping' behavior of display items; that is, the text can never render beyond the width of the display item (the real Prompt attribute has no 'width' property).
- » The prompt must be multi-line (although the Prompt attribute does support multi-line text, it requires the developer to place the carriage return character in the appropriate spot, which may not remain intact after translation. In contrast, boilerplate or a display item can automatically wrap the text).

If a display item must be used for an on-screen dynamic prompt, add Hint Text to the associated field so that the screen reader will still have access to a speakable prompt.

Note that if display items are used for this purpose, there must be code in the KEY-CLRFM trigger to repopulate the fields otherwise their 'value' will disappear if the user does a Clear Form.

## Graphics Text

Graphics text, also called boilerplate text, should be avoided. On-screen graphics text should be done with the Prompt and Label attributes wherever possible. If graphics text must be used, then an associated item must have Hint Text with similar text that can be communicated through a screen reader when that item takes focus.

## Additional Attributes

Oracle Forms thought it would be useful to have a screen reader speak additional attributes about certain items by appending the attribute after the speakable prompt.

**TABLE 3: ADDITIONAL INFORMATION SPOKEN BY SCREEN READER AFTER SPEAKABLE PROMPT**

Item	Attribute	Text Appended to Prompt
Text item	Required	Required
Text item	List of Values	List of Values
Text item	Required and List of Values	Required, List of values
Multiline item	Multiline	Multiline
Radio button	number of radio buttons in a group	n of n
Menu item	mnemonic key assigned	mnemonic x
Tab page	Tab page	Tab page

## Coding for users with physical limits

### General Rules

- » Radio buttons, check boxes and push buttons should have access keys (mnemonics) defined.
- » The keyboard tabbing sequence should follow a logical order, typically left-to-right, top-to-bottom.
- » The keyboard tabbing sequence should cycle through most if not all items in a window.
- » The user should be able to navigate between all of the blocks and windows of an application with only the keyboard.
- » Scrollbars should not be the only way to move between records or to scroll canvasses.
- » The "List Tab Pages" function (typically mapped as F2) to switch tab pages should move focus to the first navigable item on that tab page.


### Access Keys

Access keys allow operators to select or execute an item by pressing a key combination, even when that item is not the currently focused item. For example, a button labeled 'Open' with an access key of 'O' can be activated at any time by pressing Alt+O. Access keys are applicable to menu items, push buttons, checkboxes, radio buttons and tab pages. All access keys within a window should be unique, including keys used in the top level of the pull-down menu (typically F for File, E for Edit, etc.)

### Accelerators

Accelerators are keyboard shortcuts for actions that are frequently performed, for example, Ctrl+P for Print. Keyboard shortcuts allow users to bypass opening the menu by using a specific combination of keystrokes that perform the same function as a corresponding menu item. Instead of pressing Alt+F, then S, to activate menu item





File-Save, a user can just press Ctrl+S to execute the same function. Accelerator keys are desirable but not mandatory and are listed in “Keyboard Help” at runtime.

### Exceptions for Not Having Access Keys

Push Buttons, Check boxes, Tabs, Menu Items and Radio Buttons should have an access key unless:

- » They are keyboard navigable (an access key is still desirable in this case)
- » There is an excessive number of them such that deriving a unique letter would be difficult (in which case the ones with no access key must be Navigable)
- » They are not absolutely critical to the functionality of the product
- » For Checkboxes and Radio Buttons: if they are part of a multi-row block and use the Prompt, not the Label attribute, they cannot render any access key

#### *Labels*

Not every item with a 'Label' property must have an access key. If the item can be 'navigated to' (that is, it is part of the keyboard tabbing sequence), then it does not need an access key, although adding one is highly desirable as it benefits all users. Note that the Microsoft Windows User Interface Guidelines state that 'OK' and 'Cancel' buttons should not have access keys.

#### *Menu Items*

Menu items should have access keys for the top-level entries at the very least, though they are not required. The menu can always be invoked from the keyboard by pressing and releasing Alt. This will move the focus to the menu, and the arrow keys can be used to navigate through it.

#### *Iconic Buttons*

Iconic buttons (Push Buttons with the Iconic property set to Yes) cannot have an access key, therefore the functionality they invoke should be replicated elsewhere in a manner that is accessible, such as the pull-down menu.

#### *Checkboxes and Radio Buttons*

- » In a single-row block, the speakable prompt of a checkbox is almost always its Label attribute. In a multi-row block, where only a single instance of the prompt is typical, the Prompt attribute should serve as both the on-screen and speakable prompt for the item, and the Label should be null.
- » For radio buttons, use the Hint Text of the radio group to hold the speakable prompt. In a single-row block, the Label attribute of each button acts as its 'value'; in a multi-row block where only a single instance of the prompt is typical, the Prompt attribute should serve as both the on-screen prompt and 'value' for the item, and the Label should be null.
- » If the Label can be utilized, then it is very desirable to provide an access key, though not required as long as the widget is navigable. If the widget is not navigable, then using the mouse would be the only activation means, thus the access key is required.

An example of a single-row radio group:

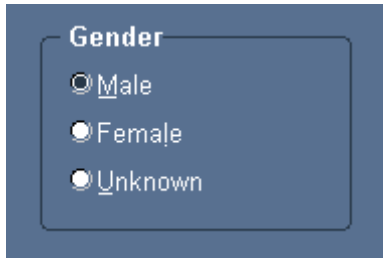


Figure 8: a portion of a screen showing a frame titled 'Gender'. Within the frame are 3 radio buttons arranged vertically, with labels 'Male', 'Female' and 'Unknown'.

- » The radio group that owns the 3 buttons has Hint Text 'Gender'.
- » The rectangle and displayed word 'Gender' are a frame widget. The title of a frame is not spoken by a screen reader.
- » 'Male', 'Female' and 'Unknown' are the Labels of the buttons, with access keys 'm', 'f', and 'u' respectively. The value of the radio group currently is 'Male'.

An example of a multi-row checkbox:



Figure 9: a portion of a screen showing 3 checkboxes in a multi-row block oriented vertically. Above them is the prompt 'Enabled'.

- » The word 'Enabled' is the Prompt of the checkbox.
- » The Label and Hint Text are both null.
- » There is no access key, so the checkbox must be navigable.

### Oracle E-Business Suite features

- » Radio groups, radio buttons, push buttons and checkboxes need to be subclassed with the appropriate FND property class. These classes specify a Pluggable Java Component (PJC) to guarantee uniqueness of access keys. At runtime, if the letter specified is already in use within the window, including the pull-down menu, a different letter will automatically be selected. This feature is most useful for applications which will be translated, where the translators may have a difficult time establishing unique access keys because they may not be operating in an environment where they can see the effect of their changes at runtime.
- » Tabs, pop-up menu items, and dynamically labeled pull-down menu items should not have access keys because they cannot be guaranteed to be unique after translation. The PJC mechanism used on other items is not possible on these.
- » An access key should be specified within the label, such as '&Print'. The separate Access Key property should not be used.
- » One default button per window should be provided, where that function is the most likely for the user to perform. Always provide a default button in a modal window (typically "OK"). All push buttons should have an access key except "OK" and "Cancel" within dialog windows. "OK" must have an access key if it is not the default button, and "Cancel" must have an access key if it does not perform the same function as "Close Window."

## Blocks

Every block must be navigable with the keyboard. Typically the navigation between blocks is controlled with the Next Navigation Block and Previous Navigation Block properties, and/or the KEY-NEXT-BLOCK and KEY-PREVIOUS-BLOCK triggers.

## Tab Pages

The WHEN-TAB-PAGE-CHANGED trigger needs to be coded properly so that focus moves to the first navigable item when the user presses "List Tab Pages" function (typically mapped as F2) to switch tab pages. The cursor does not automatically move to a different item. To move the cursor, there must be code to include a GO\_ITEM statement in the WHEN-TAB-PAGE-CHANGED trigger. This is intended functionality to allow users to view other tab pages without navigating the cursor and therefore causing item navigation and validation.

Sample 'WHEN-TAB-PAGE-CHANGED' trigger at a form level:

```
DECLARE
  tp_name VARCHAR2(30);
  tp_label VARCHAR2(30);

BEGIN
  tp_name := :SYSTEM.TAB_NEW_PAGE;
  tp_label := GET_TAB_PAGE_PROPERTY(tp_nm, label);

  IF tp_lbl = 'NAME' THEN
    GO_ITEM('FIRST_NAME'); -- or GO_BLOCK(...)
  ELSIF tp_lbl = 'ADDRESS' THEN
    GO_ITEM('ADDRESS_LINE_1');
  END IF;
END;
```

## Menus

Include the 'Window' menu entry when building a custom menu; by default the Window menu entry is not included. This entry allows a user to switch between currently open windows within the application from the keyboard.

### Coding for users with low vision

#### General Rules

- » Color-coding should never be the only means of conveying information or indicating an action.
- » Colors should not be hard-coded.
- » If fields need to be color-coded, either allow the user to turn the feature off or allow them to select from a range of colors.

#### The 'automatic' color

Oracle Forms provides a special color named 'automatic', which is essentially a 'do-the-right-thing' attribute. For example, when applied to the text and background colors of a Text Item, the text will be black and the background color will be white, when running with the Oracle Look and Feel. That same item, running in the Generic Look and Feel, may have different colors depending on how the system colors are currently set up. This flexibility allows each user to control colors so they can account for visual impairments they may have.

Note that the Oracle Forms Layout Editor does not fully recognize the 'automatic' color; it may render items with unexpected colors at design time. Only the runtime Java engine is able to completely interpret and apply this color.

## Oracle E-Business Suite features

Oracle E-Business Suite follows all of the guidelines mentioned above in this paper, but has additional requirements and features. These arise from the need to translate to multiple languages, and additional accessibility support built into the Oracle Application Object Library (FND) product. If you are coding extensions to Oracle E-Business Suite, you should follow these rules in addition to the prior material. This material is covered in further depth in the Oracle Applications Developer's Guide (part number A75545-03) for Release 11i and Oracle E-Business Suite Developer's Guide (part number E12897-04) for Release 12.1.

## Testing

To aid with testing, Oracle E-Business Suite includes a 'screen reader simulation' mode. By setting environment variable `ACCESS_READER_MODE` to the value `'ERROR_REPORTING'` before starting the web listener, the system will automatically raise an error message when focus moves to an item that does not have a speakable prompt. This mode allows validation of each form without actually having to run with a real screen reader installed, although it is not intended to be a complete substitute for such testing. This feature relies on the form-level `WHEN-NEW-ITEM-INSTANCE` trigger firing on each field, which is part of the Oracle Applications Developers Guide standards.

The Oracle Forms generic code for Actions and Values LOV and screen reader simulation mode is available upon request.

## Versions and known issues

Recommendations:

- » Microsoft Windows operating system
- » Oracle Forms Developer 6i (Patchset 5) and greater
- » Microsoft Internet Explorer 5.5 and greater browser
- » run Forms with `separateFrame=true`
- » Java Runtime Environment (JRE) Native Plug-in
- » Java Access Bridge 2.0.2 and greater

*Note:*

- » Dolphin Oceanic's Supernova Professional 8.03 is minimum version needed
- » Oracle E-Business Suite only supports `separateframe=true` where the Forms applet UI is displayed in its own window. The Forms product does support embedding the applet UI in an html page using the standard applet tags to size the window but it will not work with screen readers.
- » Oracle is working closely with assistive technology vendors to resolve accessibility issues identified to date. Please consult the My Oracle Support site at <http://support.oracle.com/> for updates to these issues.

## Conclusion

Oracle Forms provides all of the attributes necessary to build applications that are accessible to a wide variety of users. In most cases, normal Forms coding techniques would already produce the proper result; only in some exceptional cases are extra steps needed to code an accessible product. And in almost all cases, any feature added specifically for accessibility also benefits all other users.







**Oracle Corporation, World Headquarters**

500 Oracle Parkway  
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**

Phone: +1.650.506.7000  
Fax: +1.650.506.7200

CONNECT WITH US

-  [blogs.oracle.com/oracle](https://blogs.oracle.com/oracle)
-  [facebook.com/oracle](https://facebook.com/oracle)
-  [twitter.com/oracle](https://twitter.com/oracle)
-  [oracle.com](https://oracle.com)

**Integrated Cloud Applications & Platform Services**

Copyright © 2016, 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0116

Accessibility in Oracle Forms Applications  
August 2016  
Authors: Maxine Zasowski and Peter Wallack