Oracle Forms 14c New Features

January, 2025, Version [1.0]

Purpose statement

This document provides an overview of features and enhancements included in release 14.1.2.0. It is intended solely to help you assess the business benefits of upgrading to 14.1.2.0 and planning for the implementation and upgrade of the product features described.

Disclaimer

Table of contents

Introduction

This document is intended to outline some of the many new features found in Oracle Forms 14c (14.1.2). This document alone does not represent a complete collection of all the new features and enhancements introduced into this release. It is assumed that readers will be using the latest product version and therefore can take advantage of any feature discussed.

This high-level document is only for reference and an introduction to some of the new features. Complete details are available in the product documentation, including the Form Builder Help.

For a more information and usage details on these and other new features, refer to the Oracle Forms Documentation Library and Oracle Form Builder Help.

Runtime Features

User Interface

The level of customization available to the Forms user interface has remained mostly the same for almost 20 years. The need for more low code or no code customization options has been long overdue. This new functionality is a significant part of the path toward modernizing an application.

The following illustrates various item types and other Forms components and the improvements they received. More information about the functionality, how to use it, restrictions, and much more can be found in the Form Builder Help.

Text Items

Text Items, also referred to as Edit Items are fields used for end-users to enter and edit data. Historically, these items were represented as a rectangular box with a cursor illustrating the insertion point when the user enters text. Although the font styling and field color can be changed, this was mostly the extent of customization available. It is possible to introduce a background pattern to this item type, but this often was considered a distraction and made reading the text somewhat difficult.

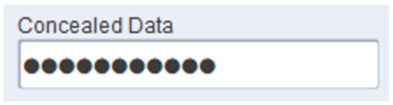
A new set of attributes has been introduced in this release, which allows for many different styles. Here are some examples of the enhancements.

Concealed Data

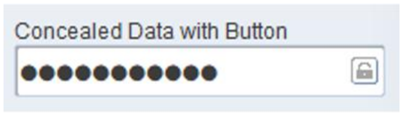
Selectable concealed data characters. Now you can choose which character to be used for concealed data fields. This includes the Forms default logon and password change dialogs. To enable this feature, no code changes are required. Simply set the desired value in the Font, Image, and Style Mapping (i.e., Registry.dat).

```
        default.concealedData.character=<Unicode value or single character>
```

For example, to use large dots set the value to `\u25cf`. This presents a large circular dot and will appear like this:



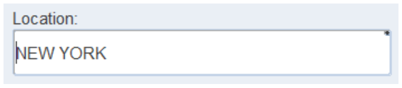
A show concealed data button can be included in the field by setting a new property, Concealed Data Button, which allows the user the ability to momentarily see the concealed data in a human readable format when pressed. When the button is released, the text will become hidden again.



List of Values Button

In case when a List Item has an associated List of Values, it may be convenient for the application developer to allow the user easy access to the LOV without the need for including any code to do it. With this feature enabled, a small button will appear on the upper right side of the field when the user enters the field. When the user moves the mouse over this button it will cause the mouse pointer to change to a hand with finger. When the user clicks on this button the associated LOV will be presented. This default LOV cannot be customized. To present a customized LOV, use the various built-ins provided for setting LOV attributes then display your LOV as you may have previously.

This feature is enabled using a new property (LOV Button) found on the Builder's Property Palette for this item.

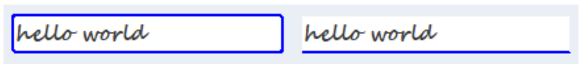


Custom Fonts

It has always been possible to use fonts found on the user's machine. However, the configuration changes needed to make that work were cumbersome. In this release, with a simple change in Registry.dat, application items configured to use Font Names that exist on the user's machine will be rendered at runtime without the need for special user-side changes (other than the font's existence). For example, set the follow parameter to "`partial`" and Forms will attempt to use the Font Name referenced at design time (or runtime). If the font is not found, the Java Font Mapping will be used and the behavior seen in earlier versions will be experienced. Valid values are; FULL, PARTIAL, NONE. Refer to the Working with Oracle Forms guide or comments in Registry.dat for details.

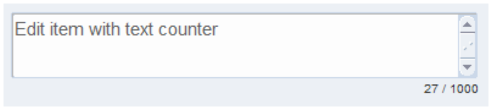
```
default.fontMap.defaultMapping=partial
```

The image below illustrates the use of custom fonts, but also custom Border Color and Sides, which is explained later in this document.



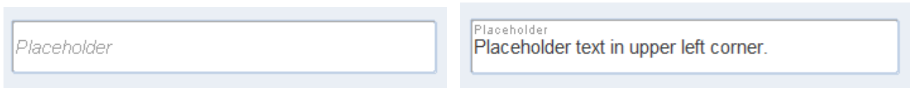
Character Counter

To expose a character counter for a text item, enable the `Display Character Counter` property. A counter will appear near the lower right side of the item when the field has focus. It will disappear when the focus leaves. The value will be presented as the number of characters in the field followed by the value of the Maximum Length property. This setting can be used on multi-line or single line text items.



Placeholder and Persistent Placeholder

Placeholder and persistent placeholder are a modern approach to item prompts/labels. By using placeholder text, you can avoid using prompts, which require extra usable space outside the perimeter of the item. When the user navigates to a text item that uses feature, the placeholder text will remain visible until the user begins entering text into the field. Similarly, when all the text is removed the placeholder text will reappear. In cases, where not using a prompt are desirable, enabling Persistent Placeholder will cause the placeholder text to minimize to approximately 70% of the field's font size and move to the upper left corner. As a result, the user can continue to see the placeholder information even after text has been entered into the field. It should be noted that in order to most effectively use persistent placeholder, the item height must be larger enough to accommodate the user's entry, which will be vertically centered and the persistent placeholder text. The placeholder color can be set globally in the Registry.dat. Refer to the Working with Oracle Forms guide for details.



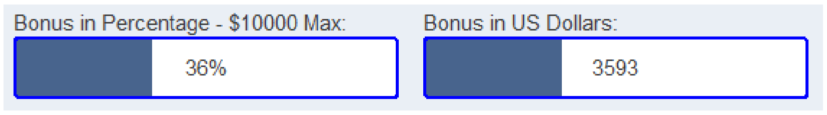
Display Items

Forms display items have traditionally been akin to a text item that was disabled. Meaning, it presented data but the user could not navigate to the item. Several new Display item properties have been introduced, which now allow this item type to be visually be presented in a variety of ways. To use these new visualizations, the item's datatype must be one of these; NUMBER, INTEGER, RNUMBER, or RINTEGER.

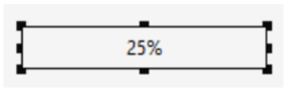
A new property, Display Item UI Style will include these options; Standard, Progress Bar, Gauge, and Half Gauge. "Standard" will be used for traditional Display items. The three new styles will be as follows:

Progress Bar

A progress bar is intended to provide a visual representation of a workflow or task status. It is represented as a rectangle with a moving solid bar within it. Progress bars can be displayed horizontally or vertically. As the value of the item increases, the amount of container rectangle filled area will also increase. The data value can be displayed as a percentage of a define maximum value relative to a defined minimum value. Both the min and max values can be set at design time or runtime. The data value displayed can also be shown as the numeric data value. However, because only whole numbers can be displayed, the value shown will be rounded to the next whole number if a fractional value is input into the item. The actual data value stored in the database will remains the complete value as store, but the presentation layer may visually alter it by rounding, as described.

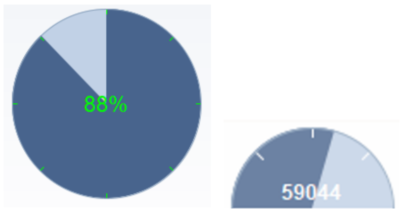


At design time, a Progress Bar will appear on the Layout Editor as a rectangle with "25%" in the center.

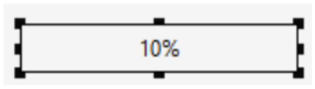


Gauge and Half Gauge

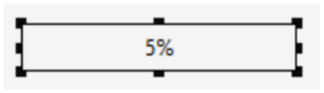
Gauges are very similar to progress bars in that they provide a visual representation of a data value. The same restrictions and limitations in progress bars apply to gauges.



At design time, a Gauge will appear on the Layout Editor as a rectangle with "10%" in the center.



At design time, a Half Gauge will appear on the Layout Editor as a rectangle with "5%" in the center.

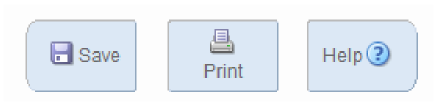


Showing the gauges as progress bars was necessary due to a limitation in the Layout Editor. Therefore, different percentages were used to help identify each of the item types.

Push Buttons

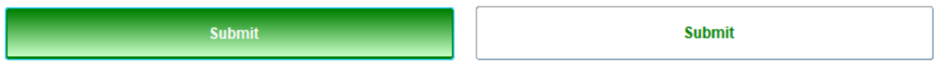
Images and Labels

Push buttons previously only supporting showing a label or an image, but not both at the same time. Showing both at the same time is now possible. The label can be positioned on any one of the four sides of the image. The label position can only be set at design time.



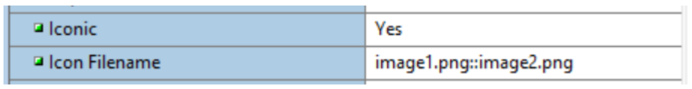
Rollover Color Swap

By setting a button's Rollover Color Swap property, the button's foreground and background colors will switch when the user moves the mouse over the button.



Rollover Image Swap

Images on buttons can be switched from one to another when the user moves the mouse over the button. When the mouse leaves the button area, the image will return to its original file. This feature is enabled by extending the existing property; `Icon Filename` by using :: (colon followed by colon) between the two image file names.

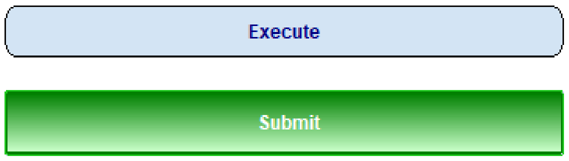


Iconic	Yes
Icon Filename	image1.png::image2.png

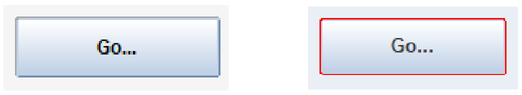
Gradient Color and Rounded Ends

Buttons can now have their left and right sides rounded or flat. The degree of the curved arc will depend on the height of the button. The shorter the button the larger the curve.

In addition to solid background colors, the use of gradient is also possible.



Applying a border color to a button can be helpful for users to more easily identify (visually) a particular button. Although the border color is set globally and cannot be changed at runtime, the colored border can be enabled and disabled at runtime. Refer to "Border Color and Sides" section below.

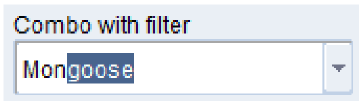


List Items

List items are used to provide the user with an easy way to access numerous possible values for a given item. The list may contain a set of fixed values or may offer a fixed list along with a text field where the user can enter an alternative value than those found in the list.

Combo Box

Combo box items allow the user to either choose a value from a set list of values or the user can enter an alternative value in the provided text field. Combo box functionality has been extended to now offer auto-completion. With auto-completion enabled, as the user enters text the combo box will internally search the contents of its list and if a match is found (based on what has been enter thus far), the full string will be presented. If the user decides to choose this value, they can simply press Tab or Enter (or use the mouse) to leave the field and the completed value will be the chosen one. If however, the string shown is not the one desired, the user can continue to type and if a different match is found it would be presented in the same way and again the user can decide to accept this suggestion or continue typing. Searching will always occur from the top of the list down with each new character entered.



Spin Box

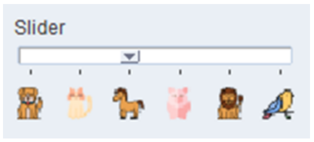
A Spin Box operates in a similar manner to a T-List. However, in the case of a Spin List if the user scrolls to the end of the list the list will move back to the beginning of the list and continue from that point. The same would occur in the reverse direction. When the beginning is reached, the user would move to the last entry in the list next.

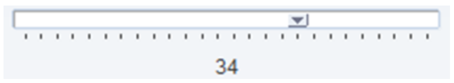
Slider

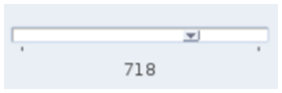
A Slider allows the user to choose a value by moving a positioning bar from left to right on a track. Like a thermometer, there are tick marks along the way that represent each value in the list. Whether or not the tick marks appear will depend on the minimum and maximum UI value is set for the item (set at design time or runtime). If the item determines that there is enough horizontal space to draw the needed number of ticks they will be included, otherwise no ticks will be shown.

The Slider's current data value will be shown (visually rounded to the next whole number) below the item if the item's height is large enough to show it. Since there is no way to programmatically hide the data value from being displayed, simply reduce the height of the item to hide it.

The Slider list type requires that the datatype be numeric (e.g., NUMBER).







All Bordered Items

Border Color and Sides

Not supported for some item types. Setting the color of the item's border is now possible by enabling one or more of these new item properties:

- `Show Border Color`
- `Rollover Border Color`
- `Specified Border Sides`

The desired value for each property can be declared in Registry.dat, which then will provide a global value for any items with those properties enabled. The default value if unset will be red (255,0,0).

```
default.border.color=<desired color in R,G,B value>

default.border.rolloverColor=<desired color in R,G,B value>
```

The `highlightColor` attribute can only be enabled programmatically.   The default, if unset will be red (255,0,0).

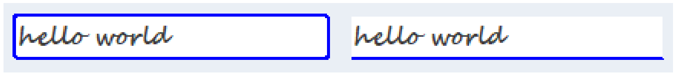
```
default.border.highlightColor=<desired color in R,G,B value>
```

To enable `highlightColor`, set the desired color as mentioned above then do the following:

```
SET_ITEM_PROPERTY ('TEXT_ITEM1', BORDER_BEVEL, HIGHLIGHTED);
```

To indicate which of the four border sides to display, enable the item's `Specified Border Sides` property then indicate with `TRUE` or `FALSE or unset (same as TRUE)`, which sides to show in the Registry.dat. For example;

```
default.border.right=false
default.border.left=true
default.border.top=false
default.border.bottom=true
```



Boolean Items

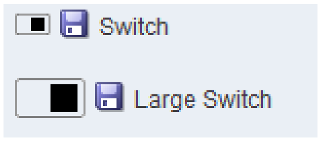
Boolean items are often represented as check boxes or radio buttons. However, in more modern designs some Booleans are represented as switches or toggle buttons.

Switch

Switches can be enabled by choosing Switch or Large Switch from Check Box or Radio Button item property UI Style. The Switch UI Style requires approximately the same vertical space used by a check box. Therefore, in cases where horizontal space is not an issue, but vertical space is limited this switch style can be used to easily replace check boxes or radio buttons without any changes to the existing positioning of items. The Large Switch style is approximately twice the height of a check box.

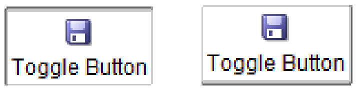
To use the switches, simply click anywhere on them to change the value (keyboard can also be used). This is the same behavior as seen with a check box or radio button. The switch is enabled (on) when the internal box is positioned to the right side. This inside handle will appear black when in the "on" position (right) and white when in the "off" position (left).

Including an image in between the object and label is also now supported.



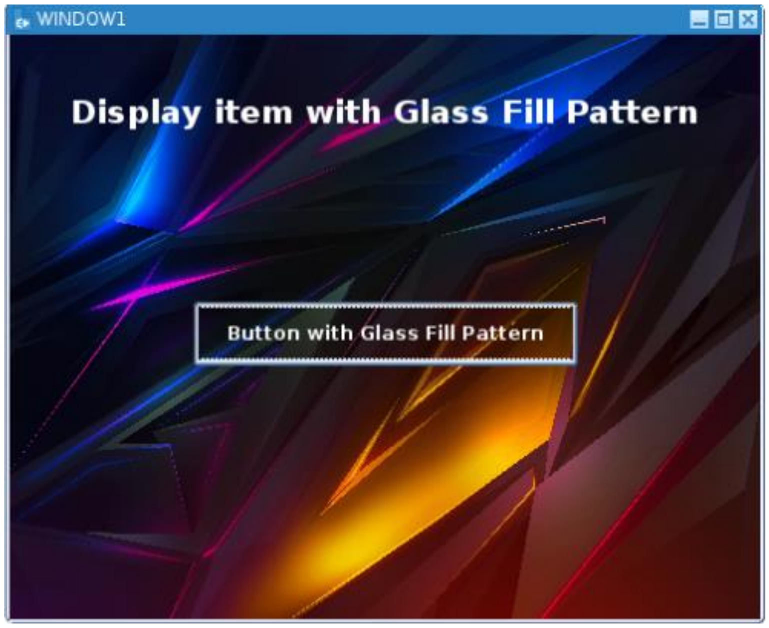
Toggle Button

To select the Toggle Button for your Boolean item, choose Toggle Button from the UI Style property of the item. The button is engaged (on) when in a depressed position and disengaged when raised in appearance (off). This style does support having a label and image, but unlike Push Buttons, the label cannot be moved relative to the image. The image will always appear above the label.



Fill Pattern Property

Glass is a new "Fill Pattern" that can be used on any item that supports having a Fill Pattern. "Glass" provides a transparent background, allowing color, images, or other objects that are behind the item to be seen through. This feature provides an easy way to have programmatically settable Display items that appear like page/screen titles. In other words, rather than using boiler text (graphic text) you can use a Display item with its Fill Pattern set to "glass" and control the text in pl/sql code.

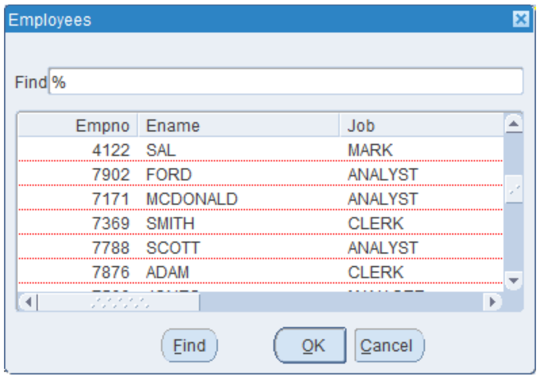

WINDOW1
Display item with Glass Fill Pattern
Button with Glass Fill Pattern

List of Values (LOV)

The Forms List of Values is a built-in compound widget used to display many values at one time. It also provides filtering functionality, allowing the user to reduce the number of visible values to only those desired. One issue with the Forms LOV is that in cases where many columns are displayed, it can sometimes be visually difficult to see across the row due to the fact that each row can give the appearance of overlapping. A new Registry.dat setting now allows administrators to enable a visible dividing line between each row, making more easily identifiable. The new setting is: lovRowLine.color and can be set to any valid RGB or OLAF value. For example:

```
default.lovRowLine.color=255,0,0
```

This will result in visible red lines appearing in the LOV, as seen in the image below.

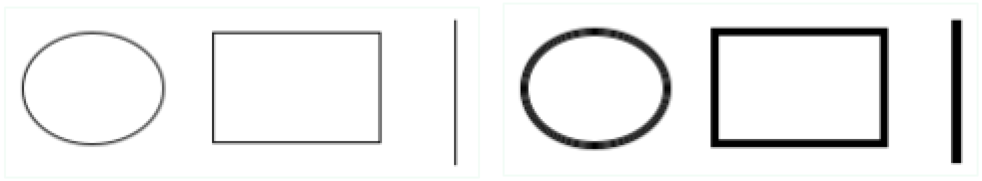


Empno	Ename	Job
4122	SAL	MARK
7902	FORD	ANALYST
7171	MCDONALD	ANALYST
7369	SMITH	CLERK
7788	SCOTT	ANALYST
7876	ADAM	CLERK

Graphic Items

Line Width

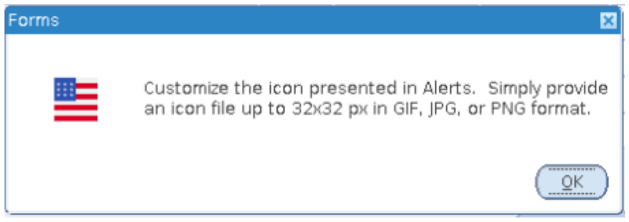
In earlier versions, the `line width` property of graphic items was ignored when using the Oracle LookandFeel. Impacted graphic types included; Rectangle, Rounded Rectangle, Ellipse, Polygon, Freehand, Polyline, Arc, and Line. By setting the `honorLineWidth` Web Configuration setting to TRUE, line width will now be honored.



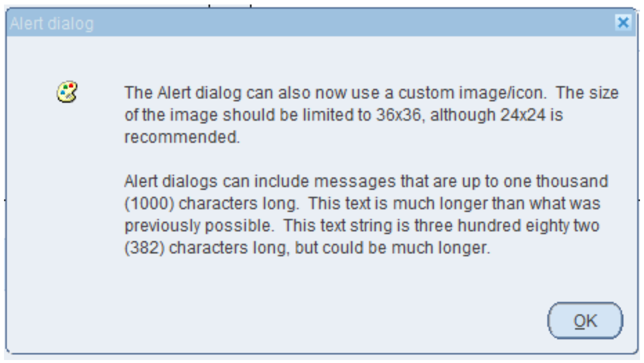
Alerts

Custom Alerts support up to 1000 characters, an increase from the previous limit of 200. Custom images can now also be used by setting the Alert property Icon Filename in the Builder or ICON_NAME at runtime. For example,

```
SET_ALERT_PROPERTY (‘ALERT1’, ICON_NAME, ‘my32x32image.png’);
```



Customize the icon presented in Alerts. Simply provide an icon file up to 32x32 px in GIF, JPG, or PNG format.

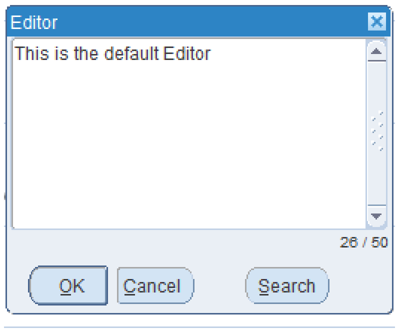


The Alert dialog can also now use a custom image/icon. The size of the image should be limited to 36x36, although 24x24 is recommended.

Alert dialogs can include messages that are up to one thousand (1000) characters long. This text is much longer than what was previously possible. This text string is three hundred eighty two (382) characters long, but could be much longer.

Editor

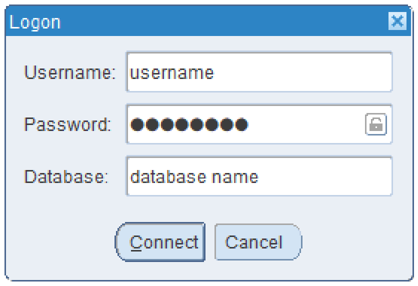
Similar to the character counter now available for text items mentioned above, the character counter has also been added to the text item default Editor and custom Editor. With a custom Editor, the counter can be optionally added, whereas with the built-in Editor it is always enabled.



Logon and Change Password Dialogs

Concealed Data Button

Both the built-in logon and change password dialogs include a "Show data" button. This button allows the user to expose the concealed password in human readable text. When the button is released, the text is concealed again. To not include this button, set the environment variable `FORMS_OBFUSCATE_CONCEALED_DATA=1` Setting this variable to 1 will result in all/any occurrences of the concealed data button not being displayed, including those explicitly enabled on text items within the applications.



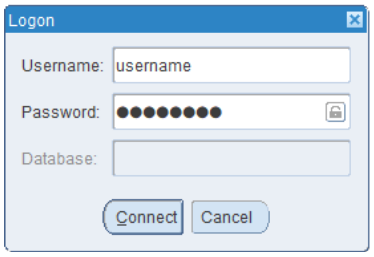
Database Name Field

To disable the Database name field in the logon dialog, use:

```
SET_APPLICATION_PROPERTY(DATABASE_NAME_ENABLED, PROPERTY_FALSE);
```

This code must be used in the PRE-LOGON trigger only. Using this setting assumes that the database name will be set in the Forms environment settings with TWO_TASK (for Linux/Unix servers) or LOCAL (for MS Windows servers). For example:

```
TWO_TASK=orcl
```



Canvases

Stacked Canvas Splitter

Stacked canvases have been used for many use-cases over the years. Likely one of the more common uses is to use a stacked canvas to contain a large amount of information, but limited the amount shown at one time. The user would use scrollbars to move view to see all the information.

Using the new stacked canvas Splitter with two stacked canvases allows the application developer the ability to position one canvas on top of the other, but also exposing a “slider” handle that allows the user to drag one canvas on top of the other. The handle can be moved by the user, using the mouse or it can be moved programmatically. It can also be locked in place to prevent the user from being able to move it. This is an ideal feature for hiding less important information, but making it easily available if needed.

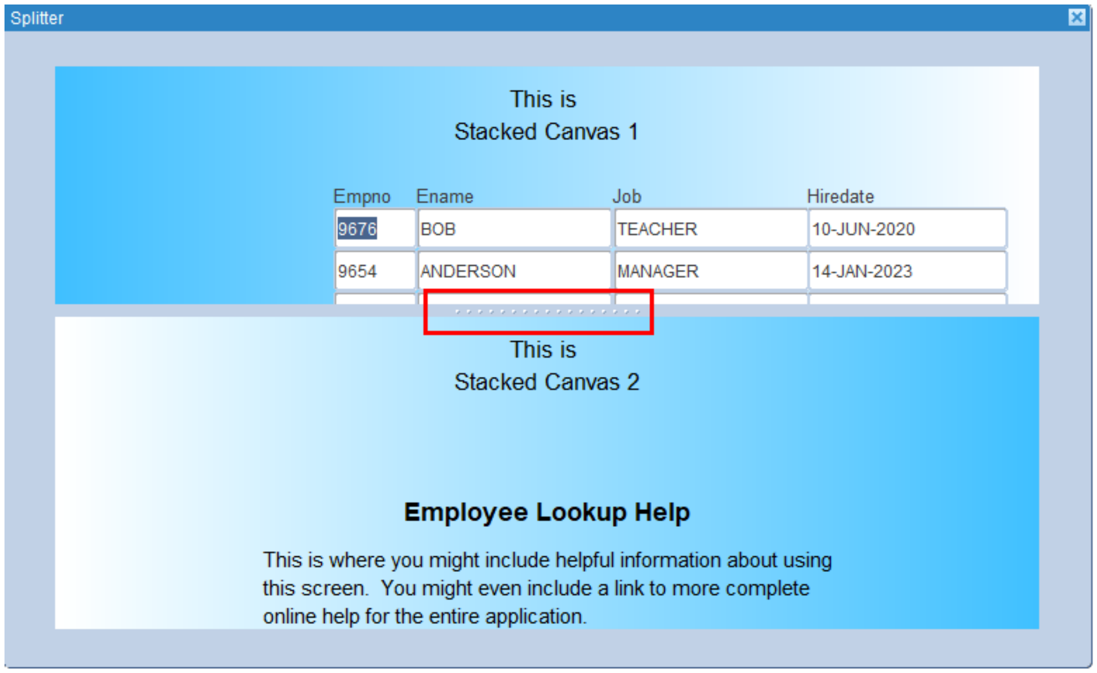
To position the Splitter programmatically, use the SET_VIEW_PROPERTY built-in as follows:

```
        SET_VIEW_PROPERTY (<CANVAS NAME>, SPLITTERBAR_POS, <POSITIOM>, <ANIMATION RATE>)
```

To disable the bar and not permit users to move it manually, use the argument SPLITTERBAR_ENABLED. For example:

```
        SET_VIEW_PROPERTY (<CANVAS NAME>, SPLITTERBAR_ENABLED, PROPERTY_FALSE);
```

The Splitter bar position can be changed even though it is disabled.



Empno	Ename	Job	Hiredate
9676	BOB	TEACHER	10-JUN-2020
9654	ANDERSON	MANAGER	14-JAN-2023

Employee Lookup Help

This is where you might include helpful information about using this screen. You might even include a link to more complete online help for the entire application.

Canvas Background Image

A canvas's background image can be set programmatically using the new canvas property, `BACKGROUND_IMAGE`. The image file can exist on the server's file system or come from a URL. To use a URL (fully qualified or relative), prefix the URL with 'URL:'. For example:

```
-- From relative URL
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE, 'URL:/forms/java/images/image.png');
-- From file system
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE, '/home/oracle/Pictures/image.png');
```

The image will be place on the canvas based on its size. Meaning that if the image size is smaller than the canvas size, it will only fill part of the canvas. However, if having the image stretched to fill the canvas is desired, set `BACKGROUND_IMAGE_ALWAYS_FILL` to `PROPERTY_TRUE` immediately before setting the image. For example:

```
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE_ALWAY_FILE, PROPERTY_TRUE);
SET_CANVAS_PROPERTY ('CANVAS1', BACKGROUND_IMAGE, '/home/oracle/Pictures/image.png');
```

Toolbar Canvas Animation

A toolbar canvas can be made visible or hidden using the SET_VIEW_PROPERTY built-in. A new argument has been added to this built-in which causes the showing or hiding occur with an animated or fluid motion. As a result, the displaying or hiding of the toolbar can be less visually abrupt. This is not supported when the toolbar is attached to the MDI window. To use this new argument, do something like the following:

```
SET_VIEW_PROPERTY  ('MYTOOLBAR', VISIBLE, PROPERTY_TRUE, 5);
```

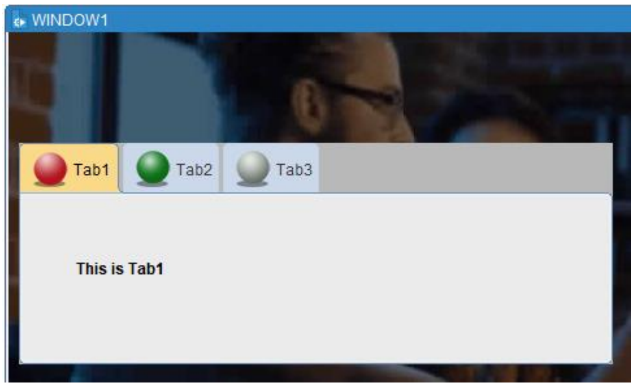
The last argument in the above example is for the animation rate, which is in the amount of time to delay between moving the next two pixels. The value is in milliseconds and valid values are from 0 to 100 with the default being 0. The exact performance of this behavior may vary depending on the user's machine performance.

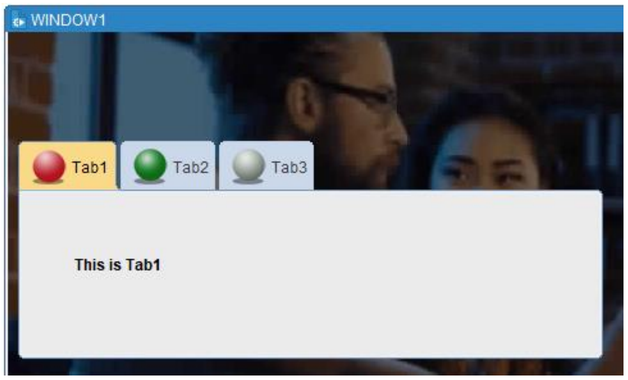
Tab Bar

Transparency

Although it may sometime not be so obvious, a Forms tabbed canvas includes a tab bar that extends the width (or height for vertical orientation) tab canvases. In the leftmost image below, notice the tab bar can be seen across the width of the tabbed canvas. This is the default behavior. However, by setting `tabBar.transparent` to TRUE the bar will become transparent and the contents behind the bar visible, as seen in the right-side image. For example:

```
default.tabBar.transparent=true
```





Selected Tab Color

Clearly identifying the currently selected tab can sometimes be difficult. By setting the background and/or foreground color of the selected and/or unselected tabs in Registry.dat, it can be much clearer to identify the current tab. Valid values will be colors identified with an RGB value or valid OLAF colors. For example, the above two images show the current tab with an orange background, while the non-current (unselected) tabs use the default values. Here are the settings for that appearance:

```
default.tabLabel.selectedFgColor=
default.tabLabel.unselectedFgColor=
default.tabLabel.selectedBgColor=OLAF2
default.tabLabel.unselectedBgColor=
```

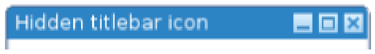
Window

Hide MDI Titlebar

The Web Configuration parameter `showMDITitleBar` provides the ability to hide/show the Forms parent window title bar. This can be helpful if a kiosk-like mode is desired. For example, if the application's parent window (`FORMS_MDI_WINDOW`) is programmatically maximized at startup and the Web Configuration include these; `showMDITitleBar=false` and `alwaysOnTop=true` the result will be a near-kiosk mode presentation. Meaning, nothing will be visible on the user's display other than the Forms application.

Titlebar Icon

The form window titlebar icon can be hidden by setting the Icon File name to the string `<none>`



AlwaysOnTop

The `alwaysOnTop` Web Configuration parameter can be used to force the application's MDI window to remain on top of all other applications running on the user's desktop. This can also be set at runtime with `SET_WINDOW_PROPERTY` and the property `ALWAYS_ON_TOP`.

```
SET_WINDOW_PROPERTY (FORMS_MDI_WINDOW, ALWAYS_ON_TOP, PROPERTY_TRUE);
```

Operating system windows like the Windows Task Manager may have priority over the Forms application window if it is also set to alwaysontop.

Window Centering

Both the form window and the application MDI windows can be centered on the display. Use the `SET_WINDOW_PROPERTY` built-in to programmatically set the window to center with the new property `WINDOW_CENTERED`. For the form window, you can alternatively set the `Center on Startup` property on the Property Palette. The MDI can alternatively be set to center by setting the Web Configuration setting `centerOnStartup=true`

```
SET_WINDOW_PROPERTY ('WINDOW1', WINDOW_CENTERED, PROPERTY_TRUE);

SET_WINDOW_PROPERTY (FORMS_MDI_WINDOW, WINDOW_CENTERED, PROPERTY_TRUE);
```

MDI Windows Resize Event

Being able to know when/if the MDI window has been resized can sometime be necessary. Because it is technically not possible to monitor the MDI window resizing from the `WHEN-WINDOW-RESIZE` trigger, it is now possible to monitor the MDI window for resizing using the `SYSTEM_MDI_WINDOW_RESIZE` in a System Event. To accomplish this, create a System Event `WHEN-EVENT-RAISED` trigger for this event with the desired code. Example:

```
DECLARE
    time VARCHAR2(20);
BEGIN
    If :SYSTEM.LAST_EVENT = 'SYSTEM_MDI_WINDOW_RESIZE' Then
           time := :System.Current_Datetime;
           message ('Resized at: ' || SUBSTR( time, instr(time,' ')), status);
    End If;
END;
```

Hide Window Menu Item

It is not uncommon to prefer not showing the "Window" menu item when not using a menu. By setting the Web Configuration `parameter hideWindowMenuItem=TRUE` the "Window" menu item will not be display IF the module does not specify a menu. This setting will retain the logo if one is used.

Canvas Window Scrollbars

Vertical and horizontal window scrollbars are typically controlled automatically. However, by setting the `SHOW_HSCROLLBAR` or `SHOW_VSCROLLBAR` property `of SET_WINDOW_PROPERTY`, the application developer can control whether these scrollbars are displayed. This only applies to the canvas window scrollbars and not the MDI scrollbars.

```
SET_WINDOW_PROPERTY (‘WINDOW1’, SHOW_HSCROLLBAR, PROPERTY_FALSE);
```

The value can also be set to `AUTO_DISPLAY` which will return control back to Forms. However, it may be possible that setting this to `AUTO_DISPLAY` may result in cases where the scrollbars are displayed when believed to not be necessary.

Default Menu & Smartbar

Default Smartbar icons have been updated. If using the old icons is preferred, set `default.icons.style=legacy` in Registry.dat (Font and Icon Mapping).

Page Setup

A Page Setup menu and Smartbar entry has been introduced to work in tandem with the Print option.

Smartbar Custom ColorScheme

The Smartbar color can be set when using a customColorScheme with the following Font and Icon Mapping (Registry.dat) entry:

```
colorScheme.sample.smartBar=<hex value color>
```

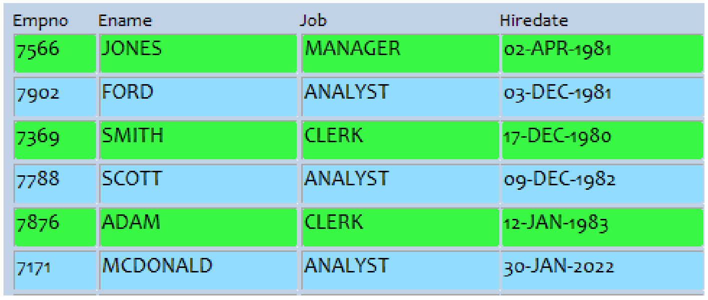
Multirecord Blocks

Row Banding

Row Banding is a feature provided in Forms 12c. This feature allowed the use of alternating colors for each row in a multirecord block. The colors were limited to those identified in the runtime colorscheme (or custom colorscheme) as the "`pinstripe1`" and "`pinstripe2`" colors. These values can be now be changed at runtime. For example, if showing the odd rows with a color in the green family and the even rows in a color in the blue family, you might do the following in your application code:

```
SET_BLOCK_PROPERTY ('EMP', PINSTRIPE1_COLOR, 'r56g248b67');
SET_BLOCK_PROPERTY ('EMP', PINSTRIPE2_COLOR, 'r146g220b255');
```

The above code example will result in something like the image below. Be sure that you have properly enabled the Row Banding property in the module. The pinstripe colors can be set at the form, block, or item level.



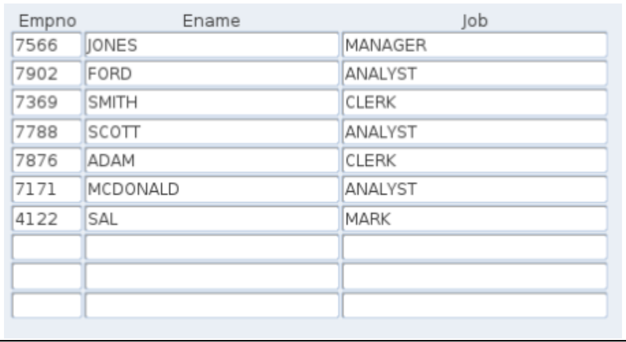
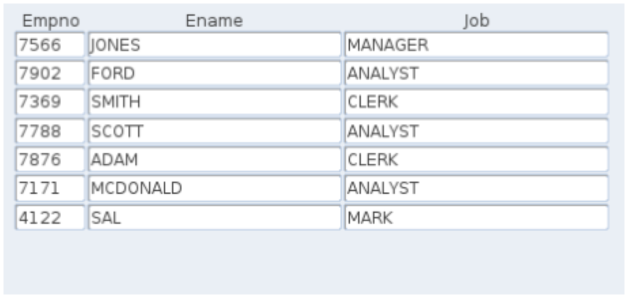
Empno	Ename	Job	Hiredate
7566	JONES	MANAGER	02-APR-1981
7902	FORD	ANALYST	03-DEC-1981
7369	SMITH	CLERK	17-DEC-1980
7788	SCOTT	ANALYST	09-DEC-1982
7876	ADAM	CLERK	12-JAN-1983
7171	MCDONALD	ANALYST	30-JAN-2022

Auto-Size Block

With Auto-Size block enabled, multi-record/row blocks will only show the number of rows necessary to present the records returned, up to the maximum number of rows identified at design-time. As a result, empty rows will not be displayed.

A new trigger; `WHEN-BLOCK-AUTOSIZED`, can be used in conjunction with Auto-Size block to control other UI components. For example, to increase or decrease the canvas and/or window height to accommodate the number of records being displayed.

Note that, the item level Number of Records Displayed property will be ignored when Auto-Size block is enabled. Also note that if the block includes a scrollbar, its height will not automatically adjust to the addition or removal of rows. However, you can change the scrollbar length using the new block property; `BLOCKSCROLLBAR_LENGTH`.

Auto-Size Not Enabled	Auto-Size Enabled
	

Empno	Ename	Job
7566	JONES	MANAGER
7902	FORD	ANALYST
7369	SMITH	CLERK
7788	SCOTT	ANALYST
7876	ADAM	CLERK
7171	MCDONALD	ANALYST
4122	SAL	MARK

Empno	Ename	Job
7566	JONES	MANAGER
7902	FORD	ANALYST
7369	SMITH	CLERK
7788	SCOTT	ANALYST
7876	ADAM	CLERK
7171	MCDONALD	ANALYST
4122	SAL	MARK

Dynamic Size Block

In some cases, the use of Auto-Size block feature might not give you the exact behavior desirable. For example, just because ten (10) records are returned it may not be desirable to display all ten. Instead, maybe showing only five (5) is preferred in this particular case. Using the new RECORDS_DISPLAYED block property you can add or remove rows as needed up to the Maximum Records Displayed defined at design-time. Example:

```
        SET_BLOCK_PROPERTY ('EMP', RECORDS_DISPLAYED, 5);
```

Note that if the block includes a scrollbar, its height will not automatically adjust to the addition or removal of rows. However, you can change the scrollbar length using the new block property; `BLOCKSCROLLBAR_LENGTH`.

Distance Between Records

In a multi-record block an item's distance between records property is most often set to 0. This results in a grid-like pattern where one row immediately follows the previous with no space between the rows. However, in some cases it may be desirable to separate the rows vertically. This may be the case if for example the field sizes are changed programmatically at runtime. Distance between records can be changed at runtime using the following example of the SET_ITEM_PROPERTY built-in. The number is the distance between records, measured in the units defined in the module's Coordinate System setting.

```
        SET_ITEM_PROPERTY ('EMPNO', RECORD_DISTANCE, <NUMBER>)
```

Tiling

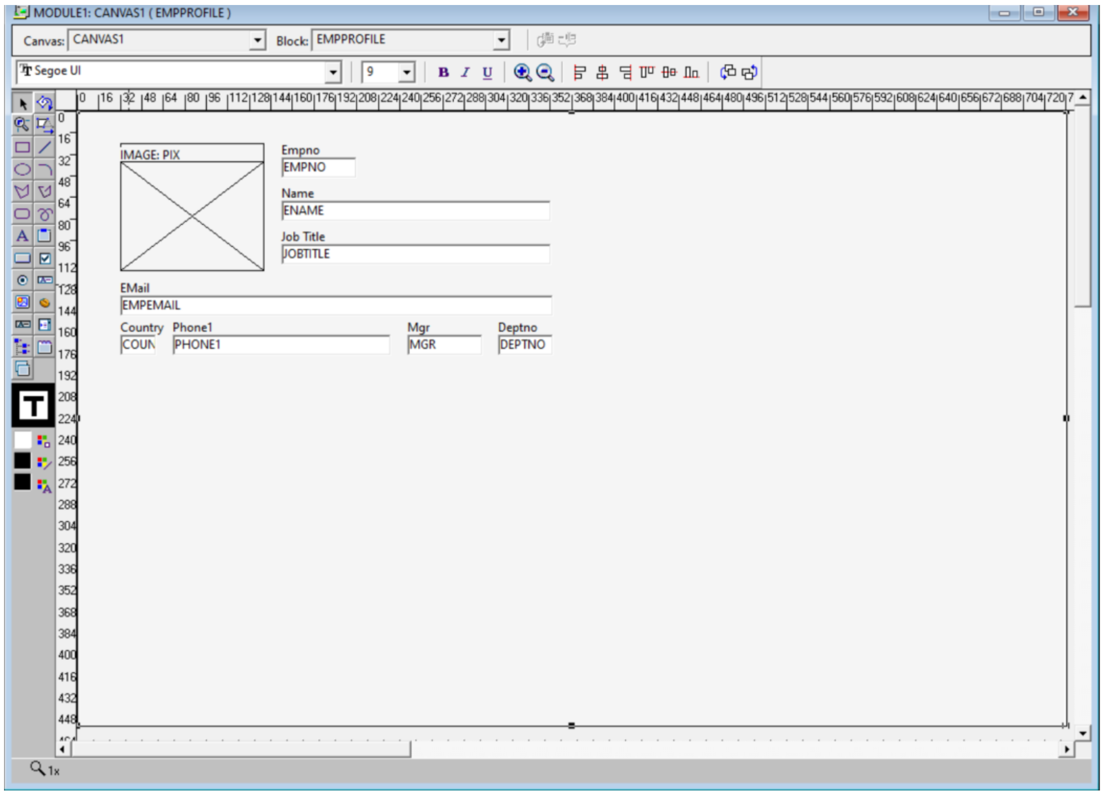
Tiling is a concept that allows each record of a multi-record block to be visually positioned or grouped together. This is also sometimes referred to as "cards". In Forms, records can be vertically or horizontally oriented. This alignment determines the navigation flow through fields. For tiling, the use of vertical orientation is recommend, but is not required.

To create a set of tiled records in a block, start with a block that has the Number of Records Displayed set to 1. If using the Layout Wizard to create a new layout from a new block, choose the Tabular layout, but set the number of records displayed to 1.

Note that frames will not be automatically created when the tiles are created. If it is desirable to have each record tile framed, it will be necessary to manually create those frames after the tiles have been created.

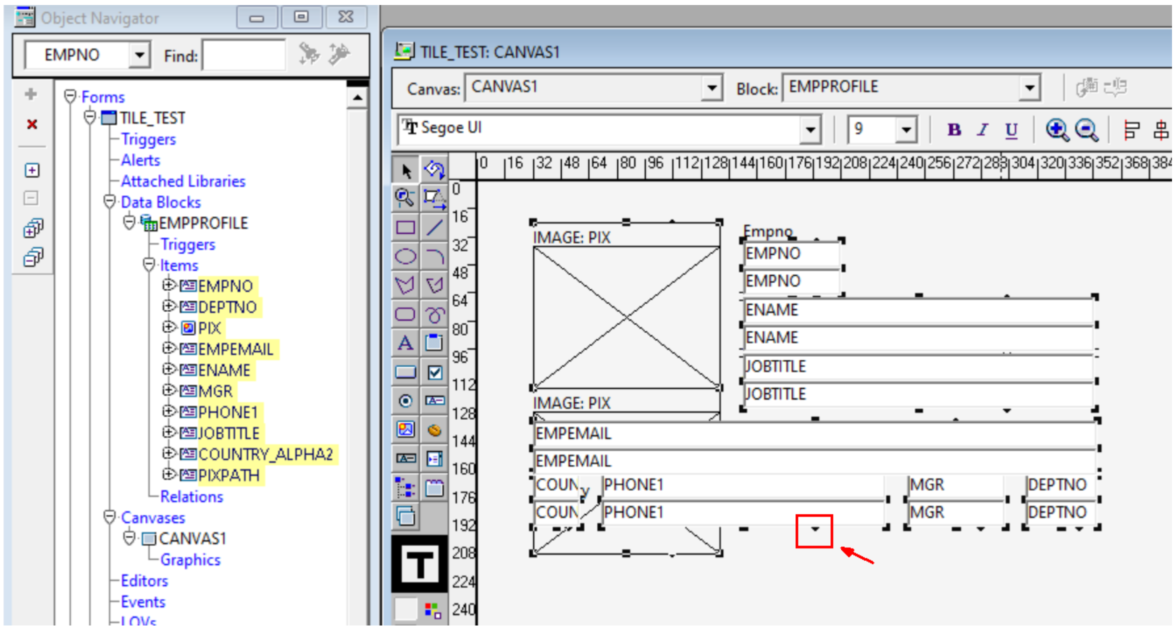
Once the single record is presented in the Layout Editor, position each item in a visual collection as desired.

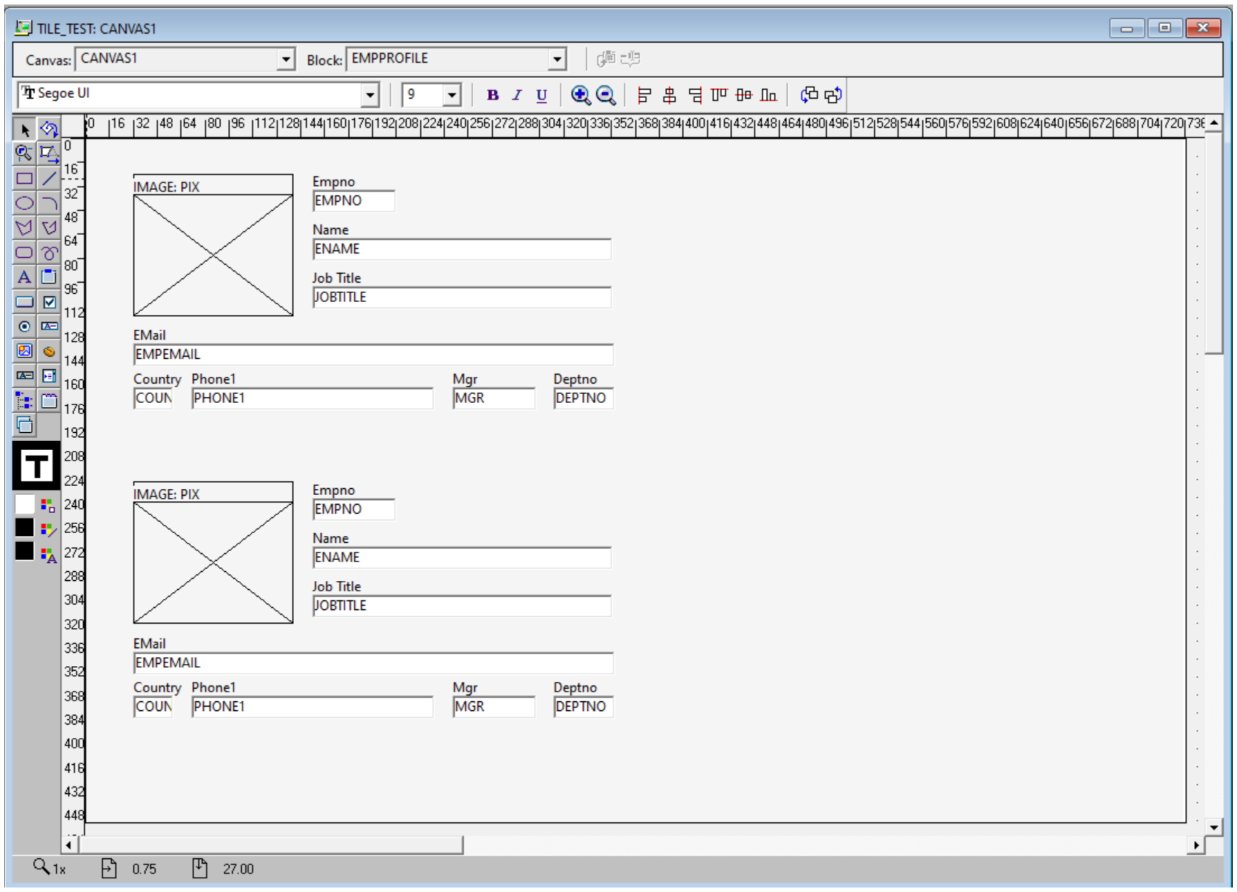
Be sure the canvas and window size are large enough to accommodate the set of tiles that will eventually be displayed. The image below represents what the initial layout may look like for a single record or tile.



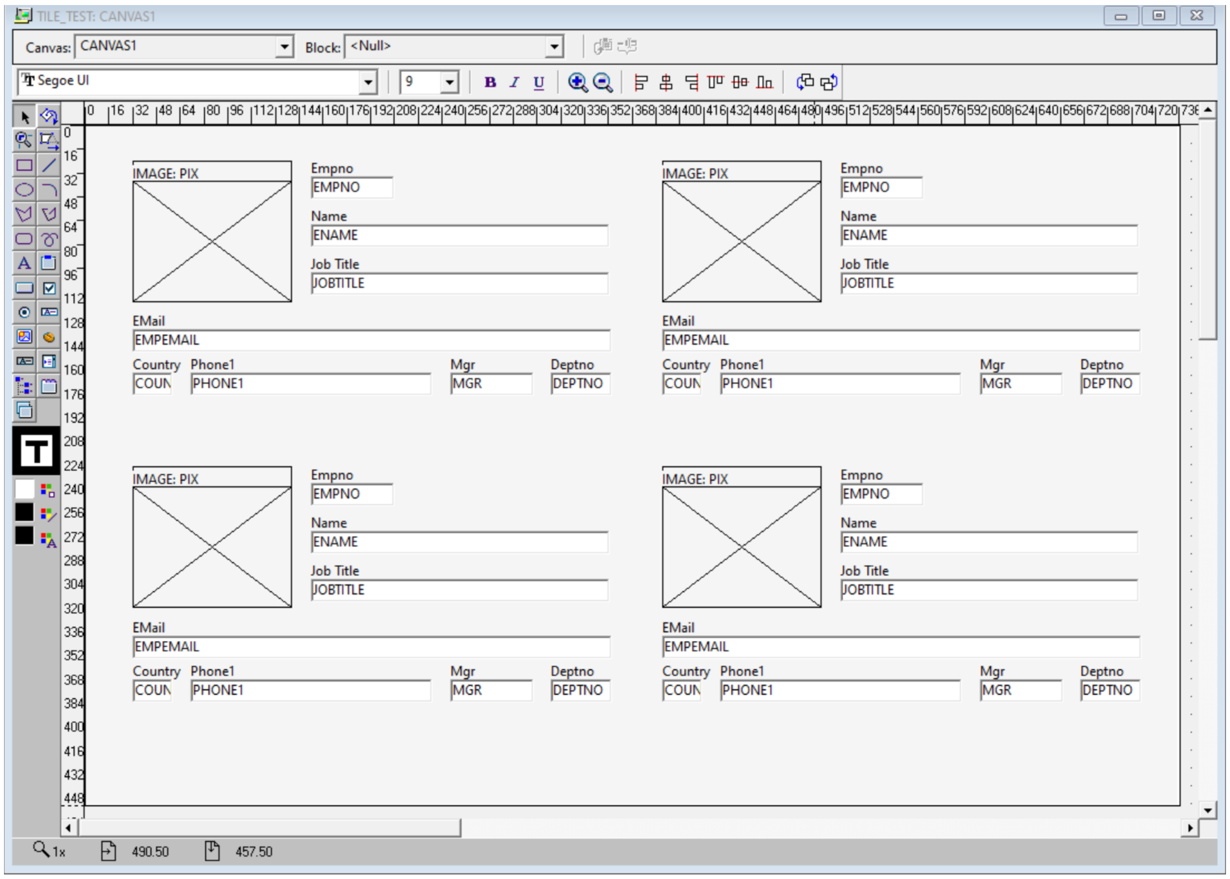
In this example, assume the desire is to show two groups of two records. Because the block is configured (by default) to have the records vertically oriented, group 1 will be the two records that appear to the left and group 2 will be the two records to the right.

Once the single record layout is satisfactory, change block's Number of Records Displayed to 2 and Records Per Tile Group to 2. This will cause a second record set of items to appear on top of the first set. In each item's Distance Between Records property, increase the value until each item is positioned below the first record and in the desired place. You can alternative select all the records items in the Object Navigator then use the Distance Between Records handle to create an initial adjustment of all items at the same time. This handle can be found at the bottom of the selection box to the right of center. It will appear as a triangle to the right of the center sizing handle. Using any of the visible triangles, use the mouse to click (and hold) and drag down to adjust the position of the second record's items. Items of the same height will move equal distance as you adjust. Items of a different height may need to have the Distance Between Records property manually adjusted in the Property Palette or using that item's Distance Between Records handle for that item to adjust using the mouse.





Now that two records have been appropriately located, set the block's Number of Records Displayed to 4. The Tile Group Distance can now be adjusted to position the remaining 2 records next to the first 2 that were created above. Increase the Tile Group Distance until this second group of records is displayed where desired.

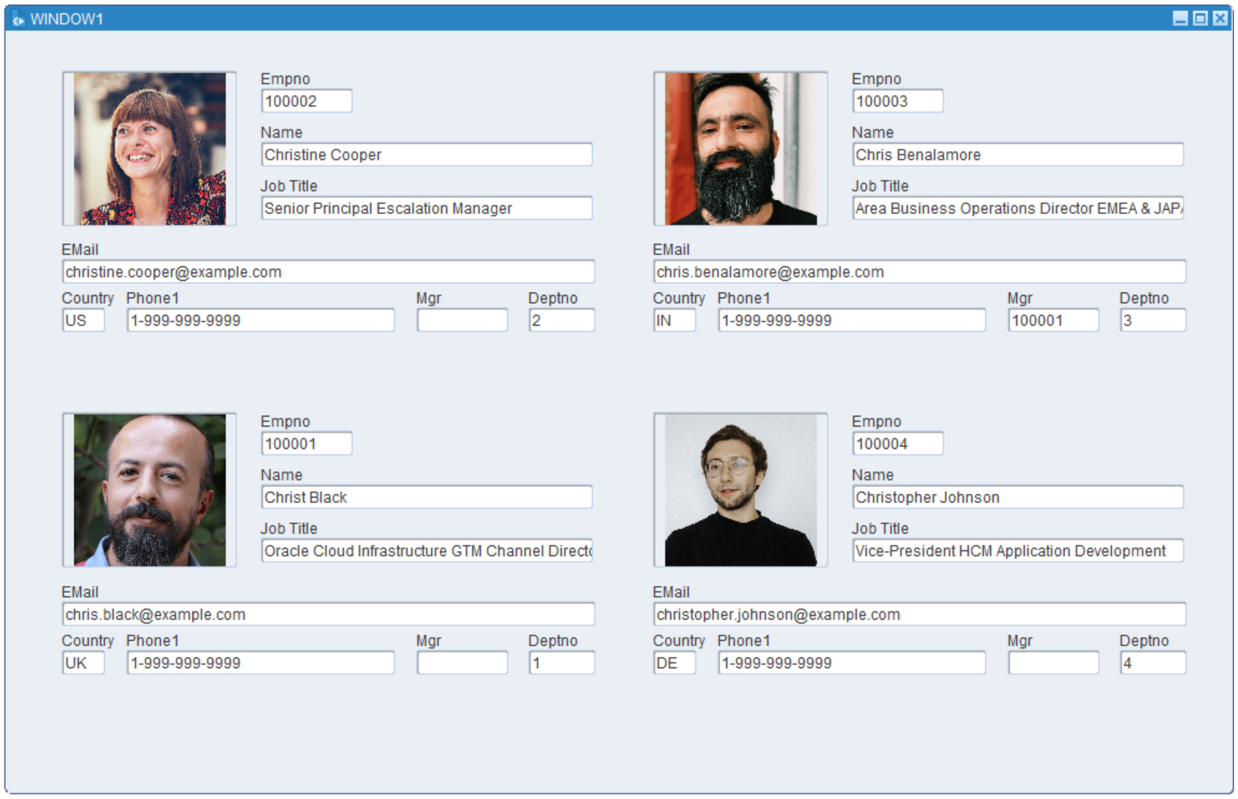

TILE_TEST: CANVAS1

Canvas: CANVAS1 Block: <Null>

Segoe UI 9

IMAGE: PIX	Empno: EMPNO; Name: ENAME; Job Title: JOBTITLE	IMAGE: PIX	Empno: EMPNO; Name: ENAME; Job Title: JOBTITLE
EMail: EMPEMAIL		EMail: EMPEMAIL	
Country: COUN; Phone1: PHONE1	Mgr: MGR; Deptno: DEPTNO	Country: COUN; Phone1: PHONE1	Mgr: MGR; Deptno: DEPTNO
IMAGE: PIX	Empno: EMPNO; Name: ENAME; Job Title: JOBTITLE	IMAGE: PIX	Empno: EMPNO; Name: ENAME; Job Title: JOBTITLE
EMail: EMPEMAIL		EMail: EMPEMAIL	
Country: COUN; Phone1: PHONE1	Mgr: MGR; Deptno: DEPTNO	Country: COUN; Phone1: PHONE1	Mgr: MGR; Deptno: DEPTNO

1x 490.50 457.50


WINDOW1

Empno	Name	Job Title	EMail	Country	Phone1	Mgr	Deptno
100002	Christine Cooper	Senior Principal Escalation Manager	christine.cooper@example.com	US	1-999-999-9999		2
100003	Chris Benalamore	Area Business Operations Director EMEA & JAP/	chris.benalamore@example.com	IN	1-999-999-9999	100001	3
100001	Christ Black	Oracle Cloud Infrastructure GTM Channel Directc	chris.black@example.com	UK	1-999-999-9999		1
100004	Christopher Johnson	Vice-President HCM Application Development	christopher.johnson@example.com	DE	1-999-999-9999		4

The result at runtime will appear something like the image shown above. Notice how there are four (4) records displayed but each record set is positioned within a tile or card-like configuration. A frame or other graphic can be used to visually separate each tile, but as mentioned that level of detail should be added after the tiles have been created because such objects will not be automatically created when the tile sets are created.

Refer to the Builder Help, specifically the page titled "About Record Tile Group" for more details information about how to use this functionality.

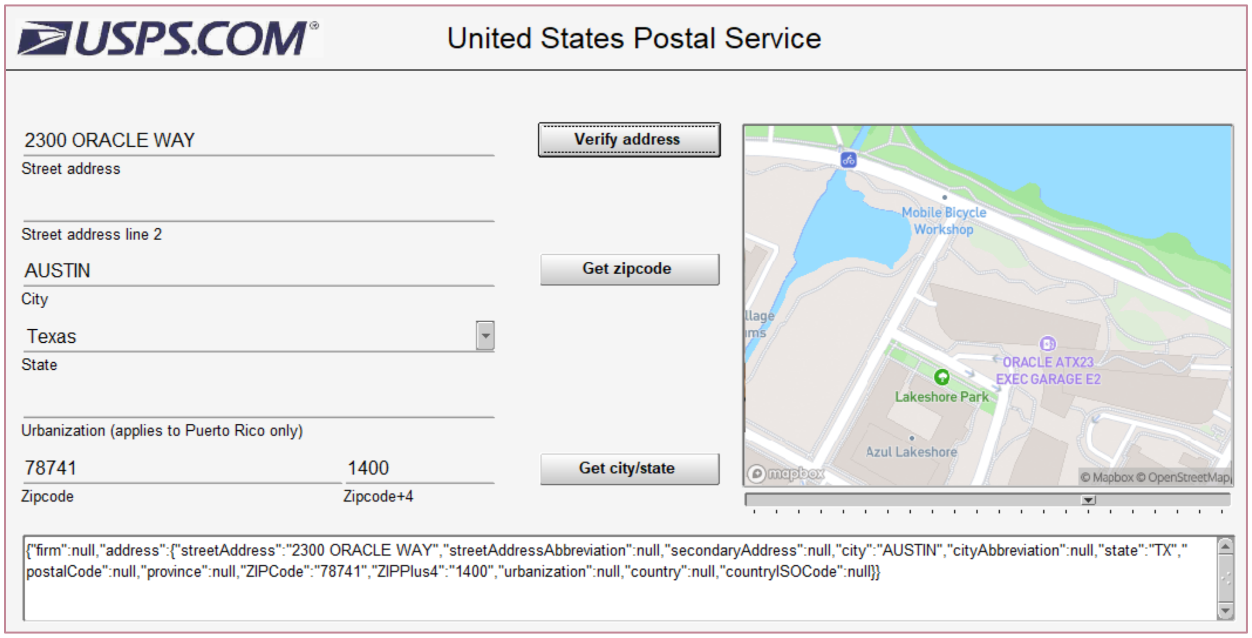
Data Access, Monitoring, and Manipulation

REST

Representational State Transfer (REST) is a way of providing interoperability between computer systems over the Internet. You can define the RESTful services to allow the querying and manipulation of data without the need for direct access to the underlying data store.

Using the new Form Builder REST Package Designer (RPD), special Forms Program Units are generated that allow the application developer to access a REST service from Forms PL/SQL without the need for creating any external code in Java or any other language. Additionally, new Forms standard packages have been introduced that will allow application developers to have granular control over the application's calls to REST services. These new built-ins will also allow for parsing and manipulating the JSON returned from a REST call. The new packages are explained in detail within the Builder Help contents.

- FSCRATCHPAD
- FJSON
- FHTTP



Street address	2300 ORACLE WAY
Street address line 2	
City	AUSTIN
State	Texas
Urbanization (applies to Puerto Rico only)	
Zipcode	78741
Zipcode+4	1400

{"firm":null,"address":{"streetAddress":"2300 ORACLE WAY","streetAddressAbbreviation":null,"secondaryAddress":null,"city":"AUSTIN","cityAbbreviation":null,"state":"TX","postalCode":null,"province":null,"ZIPCode":"78741","ZIPPlus4":"1400","urbanization":null,"country":null,"countryISOCode":null}}

Continuous Query Notification

New Forms Database Event Types for supporting Continuous Query Notification (CQN) lets an application register Forms' block queries with the database for either *object change notification* or *query result change notification*.

If a form's block query is registered for object change notification (OCN), the database notifies the application whenever a transaction changes an object (e.g., table) that the query references and commits, regardless of whether the query result changed.

If a query is registered for query result change notification (QRCN), the database notifies the application whenever a transaction changes the result of the query and commits.

By creating an appropriate Forms Database Event for OCN or QRCN and a corresponding `WHEN-EVENT-RAISED` trigger, applications can react to changes in a specific block's query. Upon receiving notification from the database that any change has been made to an object/table (OCN) or that the dataset received from a previously executed query has changed (QRCN), the `WHEN-EVENT-RAISED` trigger for the corresponding Event will fire.

This is only supported for Forms data-blocks associated with an Oracle Database object/table. This feature requires that application users have been granted "execute" permissions on `DBMS_CQ_NOTIFICATION` and `CHANGE NOTIFICATION`.

The following details can be collected regarding the notification by including the `GET_EVENT_OBJECT_PROPERTY` built-in with any one or more of the following new properties in the `WHEN-EVENT-RAISED` trigger.

- TOTAL_ROWS_AFFECTED
- TOTAL_ROWS_UPDATED
- TOTAL_ROWS_INSERTED
- TOTAL_ROWS_DELETED
- ROWS_INSERTED
- ROWS_DELETED
- ROWS_UPDATED
- TABLE_ALTERED
- TABLE_DROPPED

Refer to the Builder Help for details about what each of these properties returns.

Block Sorting

Block sorting can be used on any block (database associated or not) to sort information by an item in the same block. This sorting task will be performed entirely on the middle-tier, thereby eliminating the need to re-query the database (for database blocks). It also eliminates the need for creating sorting code in the application. With a single line of code, a block can be sorted with any of these attributes:

- ASCENDING/DESCENDING
- NULLS_FIRST/NULLS_LAST
- CASE_SENSITIVE/CASE_INSENSITIVE
- BINARY_ORDER/LINGUISTIC_ORDER

Here is an example:

```
SORT_BLOCK (:SYSTEM.CURSOR_ITEM, ASCENDING, CASE_SENSITIVE);
```

Sorting will occur with the data currently retrieved by Forms. If sorting all the records of a query immediately is desired, set the block property Query All Records to Yes. Alternatively, the sort_block command can be reissued as needed.

Supported datatypes for sorting are Number, VARCHAR2 or Date.

Miscellaneous Runtime Features

SYSTEM.CURSOR_ROW

The Forms System variable `SYSTEM.CURSOR_ROW` returns the cursor's visible row number in a multi-record block. So, regardless of the record number and its position, CURSOR_ROW will always return the current position of the cursor.

Image Copy (to clipboard)

Using the existing `COPY_REGION` built-in, it is now possible to copy images from Image items to the system's clipboard. Once on the clipboard, the image can be pasted in documents or image editors. Cut and Paste is not supported from within Forms applications. To use this functionality, navigate to the desired Image item, execute `SELECT_ALL` then `COPY_REGION`. Navigation can be accomplished by the user or programmatically. For example:

```
GO_ITEM ('IMAGE_ITEM1');
SELECT_ALL;
COPY_REGION;
```

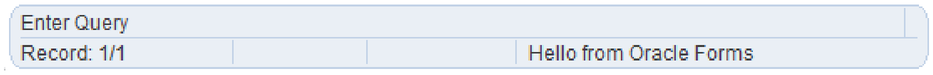
MESSAGE Built-in

By using the `MESSAGE` argument "`JAVA_CONSOLE`", message text can be sent to the Java Console or command shell (if using Forms Standalone Launcher). Example:

```
MESSAGE ('Hello from Oracle Forms', JAVA_CONSOLE);
```

By using the `MESSAGE` argument "`STATUS`", message text can be sent to the message bar line and remain persistent until this command is executed again. Example:

```
MESSAGE ('Hello from Oracle Forms', STATUS);
```


Enter Query
Record: 1/1 Hello from Oracle Forms

Get User Java Version

Sometimes it is helpful to know which Java version is being used on the user's machine. The following will return the Java version used to start the application.

```
GET_APPLICATION_PROPERTY (CLIENT_JAVA_VERSION);
```

Get Client Configuration

Sometimes it is helpful to know which Forms client configuration is being used to run the application. The following will return one of these values indicating which configuration is in use:

- STANDALONE
- WEBSTART
- EMBEDDED

Here is an example:

```
GET_APPLICATION_PROPERTY (CLIENT_DEPLOYMENT);
```

Log Modules Opened

By setting the Forms environment variable `FORMS_LOG_MODULE_OPEN=1`, modules opened at runtime will be recorded in the Forms ODL (Oracle Diagnostic Log); formsapp-diagnostic.log The entry for each module opened will appear as seen in the example below. This feature should only be used for testing purposes, as its use can result in excessive log writing to occur. It can be used in a production environment, but the log size should be monitored to ensure the file size to not get too large.

```
[2025-01-15T22:30:04.341+00:00] [WLS_FORMS] [NOTIFICATION] [FRM-91940] [oracle.forms.servlet]
[tid: 107] [userId: <anonymous>] [ecid:<EDID HERE>,0] [APP: formsapp#14.1.2] [FORMS_SESSION_ID:
WLS_FORMS.formsapp.77] [SRC_CLASS: oracle.forms.servlet.RunformProcess] [SRC_METHOD:
fromFrmwebToODL] module /home/oracle/forms/MODULE4.fmx loaded by Forms runtime process
```

WebUtil

WEBUTIL_FILE_TRANSFER.URL_TO_CLIENT

Transfer rate of URL_TO_CLIENT has been significantly improved.

WEBUTIL_FILE.FILE_MULTI_SELECTION_DIALOG

The maximum length of the string returned by FILE_MULTI_SELECTION_DIALOG has been increased from 4000 to 32767. As a result, more file names can be selected in the multi-selection file dialog.

CLIENT_TEXT_IO

The maximum line size for CLIENT_TEXT_IO has been increased from 4000 to 32767.

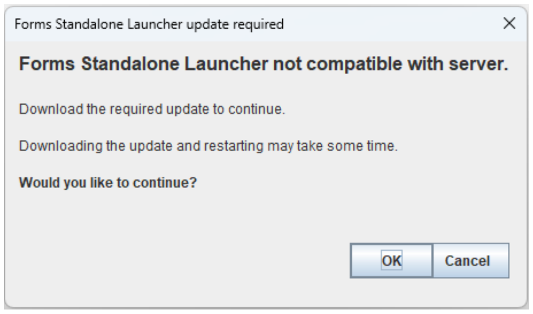
Client Configurations

Forms Standalone Launcher

Automatic Update

The Forms Standalone Launcher (frmsal.jar) is matched with the server's patch level. If the launcher and the server are not compatible with each other the user will be warned of this condition. The user will be asked whether to proceed with updating the launcher or exit. This feature is currently only supported from within the same major version. Meaning, only patched server updates can be delivered through this process. Major version upgrades must be performed manually. It is also limited to cases where the digital signature certificate has not been updated. Oracle updates its certificates approximately once every three years.

Automatic update prompting can be disabled by setting the Web Configuration setting:
`fsalEnableAutoUpdate=FALSE`


Forms Standalone Launcher update required

Forms Standalone Launcher not compatible with server.

Download the required update to continue.

Downloading the update and restarting may take some time.

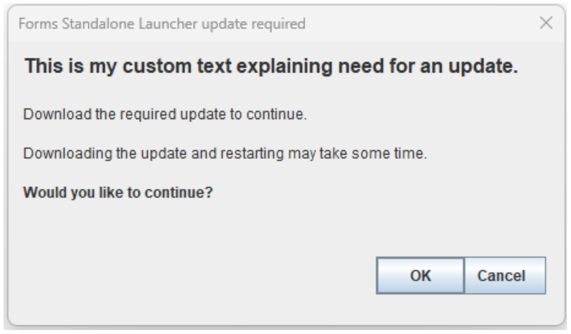
Would you like to continue?

By setting `fsalEnableAutoUpdate=FALSE, the v12 behavior will be experienced. Meaning, the administrator will need to provide the updated frmsal.jar file to all users.`

Update Dialog Text

In some cases, it may be desirable to include your own message in the dialog. The first line in the dialog can be customized with up to 128 characters of your own text. To provide custom text, set the Web Configuration parameter `fsalUpdateDialogText` and include the desired text. For example:

```
fsalUpdateDialogText= This is my custom text explaining need for an update.
```


Forms Standalone Launcher update required

This is my custom text explaining need for an update.

Download the required update to continue.

Downloading the update and restarting may take some time.

Would you like to continue?

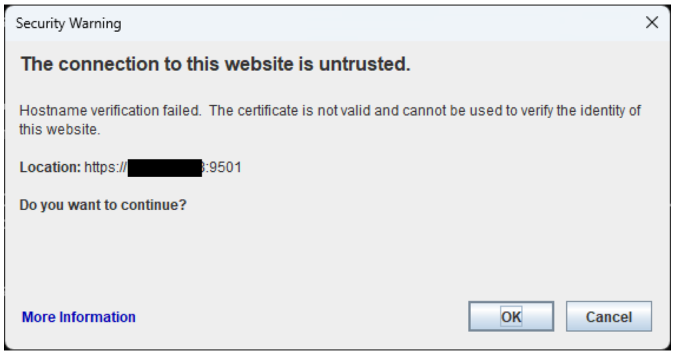
Clear Cache

A new command line argument has been introduced that can be used to clear the cached files previously downloaded by FSAL. When this argument is set to TRUE, the cache will be cleared. All other arguments will be ignored if this one is enabled. For example:

```
        java -jar frmsal.jar clearCache TRUE
```

Bypass Hostname Verification

The ability to bypass hostname verification is sometime necessary during testing. As an example, if you attempted to use the Oracle provided "demo" certificate for testing, attempts to run with SSL will fail because the certificate was not created for this purpose. By default, a dialog similar to the one below will be presented in the event the hostname cannot be verified from the certificate presented by the server.



Security Warning

The connection to this website is untrusted.

Hostname verification failed. The certificate is not valid and cannot be used to verify the identity of this website.

Location: https://[redacted]:9501

Do you want to continue?

More Information

Web Start

Automatic JNLP Deletion

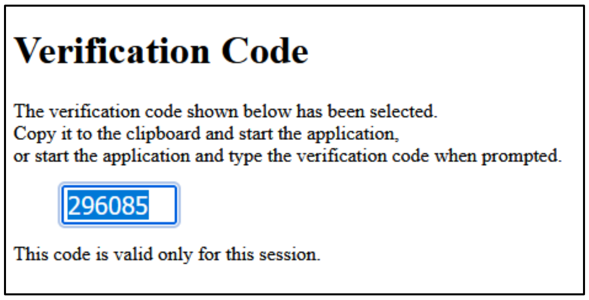
In order to ensure that the user is running the latest configuration for an application launched with Java Web Start, downloaded JNLP files are deleted automatically after they have been used. Applications should always be requested by accessing the Forms Servlet (frmservlet). Attempting to rerun and previously downloaded and used JNLP file can be risky as the administrator may have altered the application's configuration since the last time the previously downloaded JNLP file was used. If this functionality is not desired, it can be disable by editing the Forms template being used by the application (e.g. base.jnlp). The default value is "TRUE". Change this to "FALSE" to disable automatic deletion of the JNLP file. Doing this is not recommended.

```
<property name="jnlp.delete.jnlp.file" value="false"/>
```

Two Factor Authentication

In order to help ensure that users do not share downloaded JNLP files with other users and to improve the overall security of Forms applications running with Java Web Start, in addition to the Automatic JNLP Deletion feature mentioned above, Two Factor Authentication (2FA) can be enabled for Forms applications running with Java Web Start. Although this implementation is not a full 2FA solution, it will prevent downloaded JNLP files from being reused.

When enabled, the user will be presented with a Verification Code in the browser at the same time the JNLP file is downloaded. When the JNLP file is launched, a dialog requesting the Verification Code will be presented. If the correct code is not entered, the application will not start. Since the Verification Code and JNLP are both sent to the same user's machine, attempt to copy the JNLP to another machine will fail unless the code is also share. But this too can be avoided by enabling two Web Configuration settings introduced in 12.2.1.19; `jnlpMatchIP` and `jnlpTimeout`

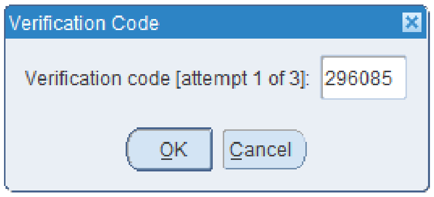


Verification Code

The verification code shown below has been selected.
Copy it to the clipboard and start the application,
or start the application and type the verification code when prompted.

296085

This code is valid only for this session.



Verification Code

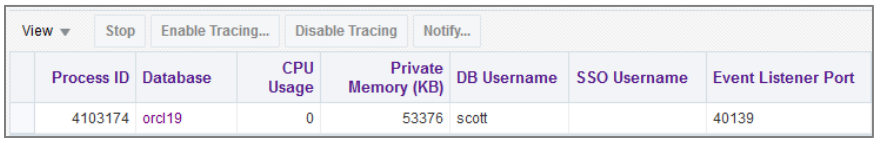
Verification code [attempt 1 of 3]: 296085

Fusion Middleware Control

User Sessions Page

Two new columns are now available on the Forms User Sessions page. SSO Username and Event Listener Port. These columns are not exposed by default, but are easily added by selecting them from the View list found at the upper left side of the table.

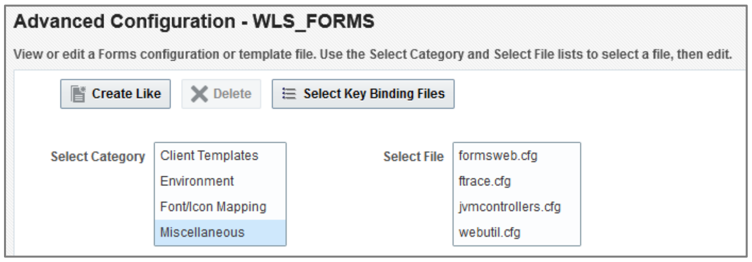
SSO Username will be populated if the session used single sign-on to authenticate. Event Listener Port is used by Advanced Queueing and Continuous Query Notification.



Process ID	Database	CPU Usage	Private Memory (KB)	DB Username	SSO Username	Event Listener Port
4103174	orcl19	0	53376	scott		40139

Advanced Configuration Page

The Advanced Configuration page provides a full text editor approach to editing configuration files. Some of the most commonly edited Forms configuration files can now be accessed from this page, thereby making it easier to make configuration changes.



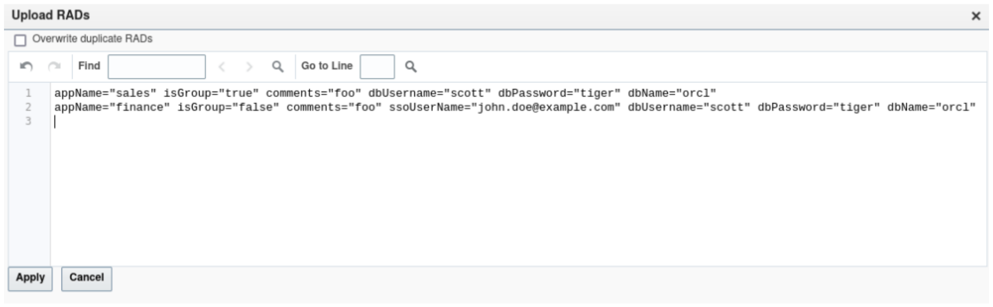
Advanced Configuration - WLS_FORMS

View or edit a Forms configuration or template file. Use the Select Category and Select File lists to select a file, then edit.

Select Category		Select File	
	Client Templates		formsweb.cfg
	Environment		ftrace.cfg
	Font/Icon Mapping		jvmcontrollers.cfg
	Miscellaneous		webutil.cfg

Remote Access Descriptors

Remote Access Descriptors (RADs) are used to aide in the process of authenticating through single sign-on. Being able to bulk upload RAD entries for many users at one time can help to ease the administrative effort of moving from one system to another. Refer to the Working with Oracle Forms Guide for information about the required syntax for bulk uploading.



```
appName="sales" isGroup="true" comments="foo" dbUsername="scott" dbPassword="tiger" dbName="orcl"
appName="finance" isGroup="false" comments="foo" ssoUserName="john.doe@example.com" dbUsername="scott" dbPassword="tiger" dbName="orcl"
```

Builder

Expose FSAL Output to Log

The ability to launch a form from the Builder is not a new concept. However, in 12c the ability to launch a currently open form with Forms Standalone Launcher was introduced. One problem with this was that shell messages like errors, warnings, and general information messages were masked. There was no way to see these messages and therefore there was no easy way to troubleshoot issues related to using the OneButtonRun Builder option with FSAL.

With a new environment variable; `FORMS_BUILDER_FSAL_LOGGING` is enabled, a log file will be created in the system temp directory. The file name will be `fsal_<random string>`. This file will contain all the details typically exposed when running FSAL directly from the command line. Valid values are as follows:

- 0 indicates no additional information should be displayed.
- 1 shows details about location from where resources are loaded.
- 2 shows details about SSL/TLS related information when certificate information is not in the FSAL TrustStore.
- 99 shows all the details.

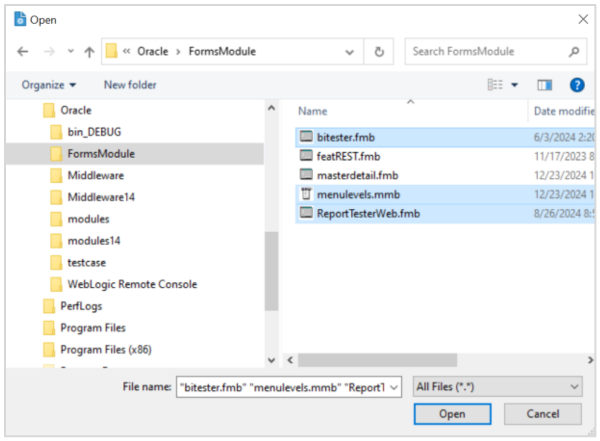
Because a new log file will be created each time FSAL is launched, enabling this feature should only be used when troubleshooting.

Improved Startup Performance

On MS Windows the Builder can sometimes appear to take a very long time to startup and become usable. This is because the Builder attempts to evaluate all fonts installed on the machine and determine which it can use and how to use them. So for machines with many fonts, startup times will be delayed. By enabling (1) the environment variable TK2_DELAY_WIDTH_CALC in the Windows Registry, the Builder will skip the font check at startup and instead only check as needed.

Multiple File Open Dialog

Being able to select and open more than one module at a time is a convenient way to begin working on the project quickly. By using the CTRL or SHIFT key along with mouse clicks, you will be able to select any number of modules to load into the Builder.

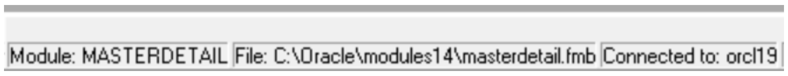


Name	Date modifie
bitester.fmb	6/3/2024 2:2
featREST.fmb	11/17/2023 8
masterdetail.fmb	12/23/2024 1
menulevels.mmb	12/23/2024 1
ReportTesterWeb.fmb	8/26/2024 8:

File name: "bitester.fmb" "menulevels.mmb" "Report1

Database Name in Message Bar

With the database name presented in the message bar, it is now easy to determine which database is being used.



Module: MASTERDETAIL | File: C:\Oracle\modules14\masterdetail.fmb | Connected to: orcl19

Form Property – Application Name

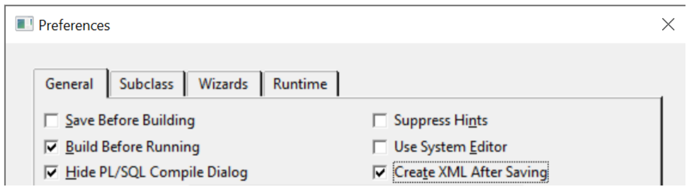
A new form level property; Application Name can be used categorize modules and later be able to identify them using the Java Development API (JDAPI). This property can only be set in the Builder and accepts any string value, although it is recommended that it be used to provide a name for the application to which this module applies. In cases where a module may be used in several applications, a comma separate list is recommended. For example: `sales, hr, inventory`

Using this property, JDAPI code can make an intelligent decision regarding whether this module should be processed and how. Forms 14c JDAPI JavaDoc can be downloaded from the Oracle Forms product home page.

Create XML After Save

The Forms module types used by the Builder are binary, non-human readable files. The non-human readable binary files make tasks like version control, code comparison, and other development and administrative tasks difficult if not impossible to achieve. The ability to convert modules to other formats like XML have been possible for a long time. However, doing so required addition steps to perform the conversion.

By setting the new Preference; `Create XML After Saving`, the Builder will attempt to create an XML version of the FMB, MMB, or OLB file opened in the Builder when saving changes. When attempting to save changes to an open module, the module's binary will be saved first then the XML Converter will be used to create an XML version of that binary module. A dialog showing the completion of this conversion will be presented. If showing this dialog is not desirable and a more silent approach is desired, set `FORMS_BUILDER_SILENT_CONVERT=0` However, understand that if performing a silent conversion, failures messages will not be seen.

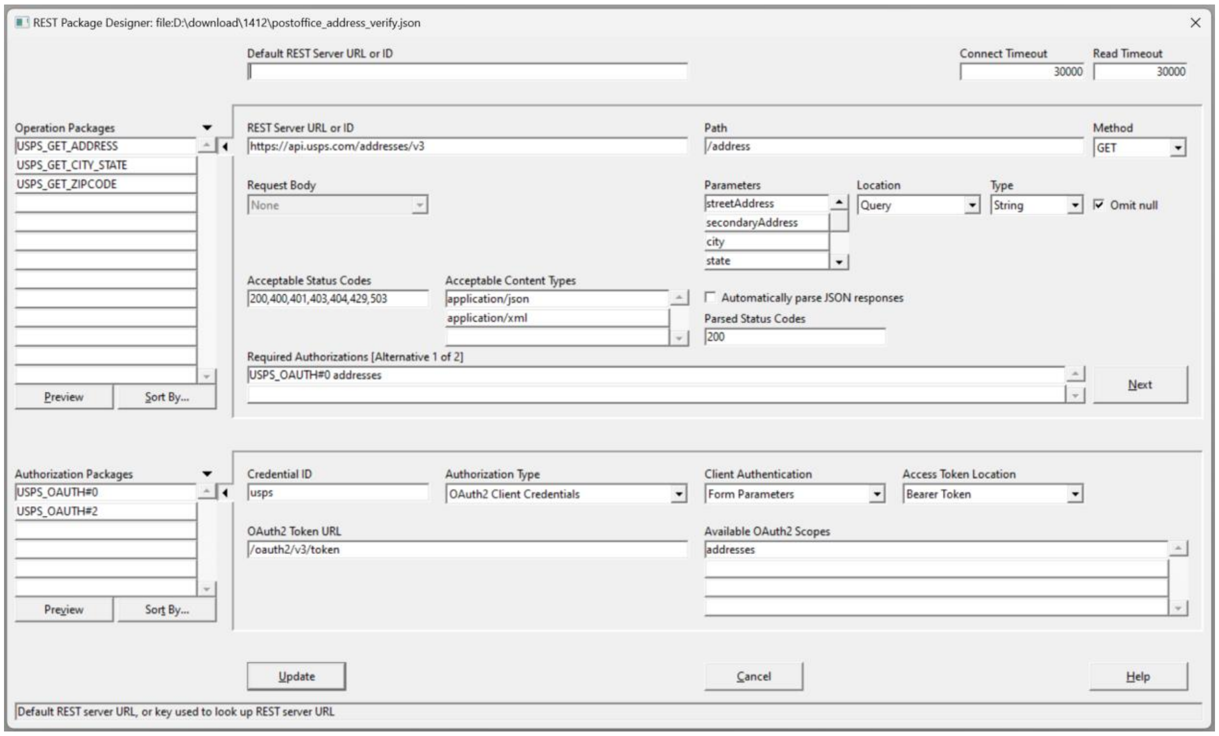


- [ ] Save Before Building
- [x] Build Before Running
- [x] Hide PL/SQL Compile Dialog
- [ ] Suppress Hints
- [ ] Use System Editor
- [x] Create XML After Saving

REST Package Designer (RPD)

The REST Package Designer or RPD, is a key component to allowing Forms developers to create application code that can make calls to REST endpoints to access and manipulate data. The RPD is used to create special Forms Program Units (PUs) intended for working with REST. These program units are created automatically based on information provided in RPD. In cases, where a REST service does not offer an OpenAPI or Swagger document, RPD can be used to manually provide details about the service. Once those details are provided, RPD will automatically create special Forms PUs, which can then be coded against. For cases where the service provides a compatible OpenAPI or Swagger document it can be used to automatically populate RPD with most of the needed information. That said, making edits in RPD is always possible regardless of how the initial information was entered (automatically or manually).

Only OpenAPI or Swagger 2.x or 3.x is supported and the document must be JSON formatted. YAML and other formats are not supported. To use other formats like YAML, a YAML to JSON converter can be used to create a compatible document. There are many conversion tools available online that can perform this task.


REST Package Designer: file:D:\download\1412\postoffice_address_verify.json

Default REST Server URL or ID:
Connect Timeout: 30000
Read Timeout: 30000

Operation Packages: USPS_GET_ADDRESS, USPS_GET_CITY_STATE, USPS_GET_ZIPCODE

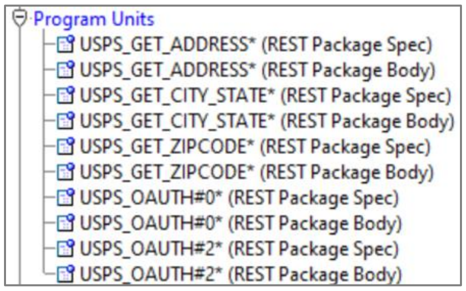
REST Server URL or ID: https://api.usps.com/addresses/v3
Path: /address
Method: GET
Request Body: None
Parameters: streetAddress, secondaryAddress, city, state
Location: Query
Type: String
Omit null: ☑
Acceptable Status Codes: 200,400,401,403,404,429,503
Acceptable Content Types: application/json, application/xml
Automatically parse JSON responses: ☐
Parsed Status Codes: 200
Required Authorizations [Alternative 1 of 2]: USPS_OAUTH#0 addresses

Authorization Packages: USPS_OAUTH#0, USPS_OAUTH#2

Credential ID: usps
Authorization Type: OAuth2 Client Credentials
Client Authentication: Form Parameters
Access Token Location: Bearer Token
OAuth2 Token URL: /oauth2/v3/token
Available OAuth2 Scopes: addresses

Default REST server URL, or key used to look up REST server URL

Once the packages are created, they will include a REST Package Spec and Body identifier. The contents of these packages should not be manually edited. They should only be edited by reentering the RPD.


Program Units
- USPS_GET_ADDRESS* (REST Package Spec)
- USPS_GET_ADDRESS* (REST Package Body)
- USPS_GET_CITY_STATE* (REST Package Spec)
- USPS_GET_CITY_STATE* (REST Package Body)
- USPS_GET_ZIPCODE* (REST Package Spec)
- USPS_GET_ZIPCODE* (REST Package Body)
- USPS_OAUTH#0* (REST Package Spec)
- USPS_OAUTH#0* (REST Package Body)
- USPS_OAUTH#2* (REST Package Spec)
- USPS_OAUTH#2* (REST Package Body)

Summary

Oracle Forms 14.1.2 comes packed with many new features. From improvements to it user interface, to its ability to easily access data from REST service calls, modernizing Forms applications has become a much easier task. This document includes many of the new features, however it does not include all. Some new features are much more subtle and didn't warrant inclusion. Be sure to refer to the References section of the Working with Oracle Forms guide as many of the new settings that can be used with features found in this document and many others are included there.

Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle