Coding Standards Changes for Forms and Libraries

ORACLE

Overview of Coding Standards Changes for Forms and Libraries

- Many changes are done for you by the Oracle Applications upgrade utility.

  You may need to make additional changes.
- You need to code new forms correctly.

Types of Windows

- Nonmodal
- Modal
- Modal with menu
- Semimodal

Nonmodal (Regular) Windows

- Nonmodal windows can access the menu and toolbar.
- Nonmodal windows use the WINDOW property class (close, move,resize, maximize, minimize)

Modal Windows

- Modal windows *cannot* access the menu or toolbar at all.
- WINDOW_DIALOG property class to disable the appropriate menu and toolbar
- As always, you must explicitly disable KEY-triggers for actions that do not apply.

Semimodal Windows

- Semimodal windows are a special case of nonmodal (regular) windows.

  They use the WINDOW property class.
- User cannot navigate to certain window without completing the semimodal window
- Avoid creating new semimodal windows.

Modal with Menu Windows

- Modal with menu windows are no longer necessary.
  - LOVs are now accessible without toolbar or menu.
  - Help buttons are now required in all modal windows.
- Do not code any new ones.

Old Modal with Menu Windows

- The WINDOW_DIALOG_WITH_MENU property class is still backwards compatible.
- Over time, recode these windows as modal windows using the WINDOW_DIALOG property class.

- Add a Help button to the recoded window.

Changes to Standards for Window Closing Behavior

- Every window must support closing using window icons.
- APP_CUSTOM.CLOSE_WINDOW must account for every window of your form (except those referenced from APPSTAND).
- Closing a modal window should be the same as if the user pressed Cancel or the equivalent.

Modal Window Closing Behavior

For each modal window:

- **The WINDOW_DIALOG property class sets Close Allowed to No.**
- **Override it and set it to Yes once you have coded the logic for the modal window into APP_CUSTOM.CLOSE_WINDOW**

Underlying Changes to Item Properties

- The Insert, Update, and Allowed attributes now automatically have colors associated with their settings.
- The field background and text colors change on a row-by-row basis (the “item instance level”).

Setting Item Properties

- APPCORE takes care of item properties at the item or instance level as appropriate.
- Be sure that you *always* use app_item_property.set_property for:
  - ENABLED
  - ENTERABLE
  - ALTERABLE
  - ALTERABLE_PLUS
- Also use it for setting most other properties.

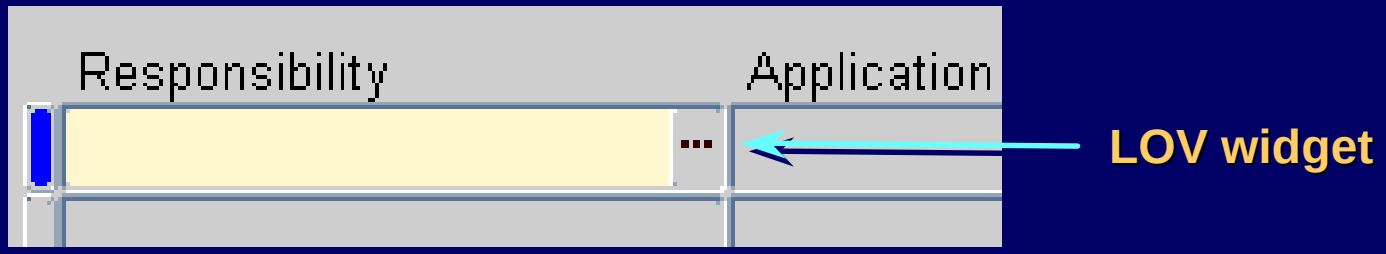
Avoid Setting Field Properties Without APPCORE

- The danger of using the native built-in set_item_instance_property is the extra level of inheritance.
- If you are using item-instance-level calls, make sure the item-level settings always allow the function.
- APPCORE routines take care of this automatically.

Lists of Values

Any text item that is editable and has an LOV associated with it gets an LOV button next to it upon focus.

- No longer need List of Values on the menu or toolbar.
- LOV access is no longer an issue in modal windows.



Dummy Lists of Values

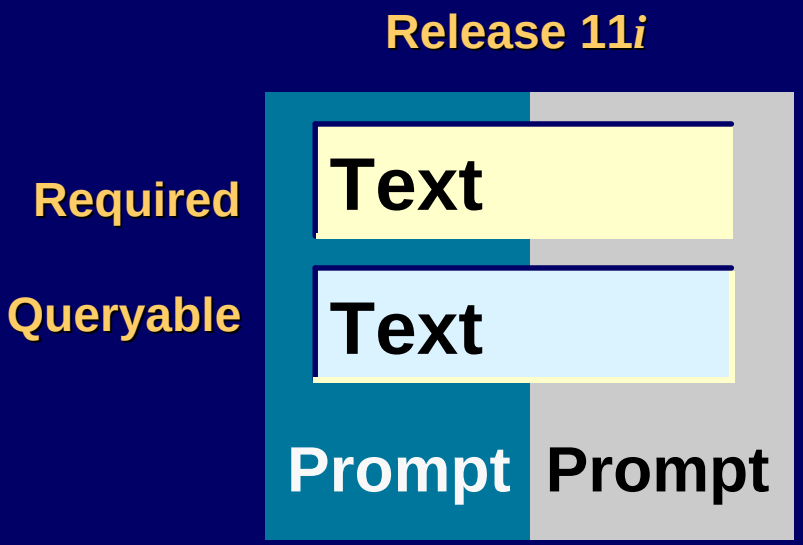
- Key flexfields and date fields must have the dummy LOV ENABLE_LIST_LAMP associated with them
- Validate from List must be set to No for these LOVs

New LOV Properties

- Automatic Position: Opens the LOV near, but not obscuring, the current field
  - SHOW_LOV will open the LOV centered on the screen
  - Overrides any (X,Y) positions that were set
- LOV property class in APPSTAND sets Automatic Position to Yes

Achieving the Oracle Look and Feel

Colors of Required and Queryable Items



Colors in the Oracle Look and Feel

- Most colors are controlled by the Oracle Applications property classes, which also set visual attributes.
- Most APPSTAND visual attributes now set the foreground and background colors to “automatic”, which means “do the right thing”.

Colors in the Oracle Look and Feel

- Non-editable items (some text items, all display items and poplists), default to black text on gray backgrounds.
- Default colors can be overridden by setting colors to something other than “automatic”.
- If you use non-APPSTAND visual attributes, you must confirm that these work as expected in custom forms.

No Need to Adjust the Color on Your Monitor

- The Layout Editor will not render widgets with the proper color.
  - The Oracle Look and Feel gets applied at runtime.
  - Only the Java code executing at runtime knows the rules.
- Do not trust, or get worried by, any colors you see in the Layout Editor.

Magic at Runtime

- Canvases assume proper color
- Boilerplate: All prompts are now attributes of fields
- Item property classes make prompts black or white depending on the canvas color
- Dynamic titles and prompts assume correct colors
- Frames have a rounded look at runtime

The Ins and Outs of Bevels

- The Oracle Look and Feel calls for all fields to have bevels.
- The APPSTAND property classes DISPLAY_ITEM and TEXT_ITEM_DISPLAY_ONLY have been modified to reflect this.
- Upgrade utility automatically moves fields with the DISPLAY_ITEM and TEXT_ITEM_DISPLAY_ONLY property classes up to nearest 0.125 inch Y value to account for the 0.02 inch offset the prior standards required.
- Bevel, Height, and Inter-record spacing have been changed.

New Property Classes for Frames

- Use frames instead of boxes or lines to achieve the Oracle Look and Feel.
- Use these property classes for frames:
  - FRAME_RECT
  - FRAME_HORZ_LINE
  - FRAME_VERT_LINE
- The Oracle Applications upgrade utility applies these to your current rectangles and lines.

New Property Classes for Fixed Boilerplate

- BP_PROMPT
- BP_TITLE

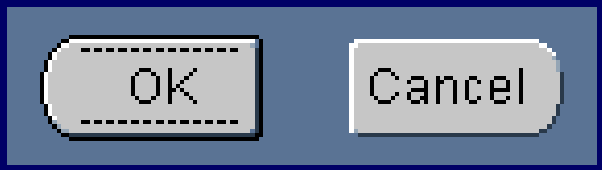
New LOV Properties

- Automatic Column Width: Ensures that the column width is not shorter than the column label

  You still must size the columns appropriately for the data being shown.
- LOV property class in APPSTAND sets Automatic Column Width to Yes

OK and Cancel Buttons

- OK and Cancel buttons should be grouped together and shown in order:



- APPSTAND objects and AOL popup windows have changed to this sequence
- Changing your forms to match is optional but recommended for a consistent interface.

Help Buttons

Modal windows should also include a Help button.

- The Help button is not part of the OK/Cancel group.
- The Help button calls window-level help.
- The WHEN-BUTTON-PRESSED trigger does:
  do_key('KEY-HELP');

FAQ

When you have to regenerate (recreate the FMX) your Apps forms or when your Apps forms are regenerated, some are true some are not:

After an operating system upgrade? False

After a database upgrade? False

After applying a Developer patch? False.

After applying an Apps patch? True, but only if there is a g driver involved.

FAQ

Forms will attempt to resolve a program unit call in the following sequence:

Program Units node

Forms Library

Database

FAQ

- How do you get the latest package release?

```
SQL> select name,type,text from user_source where
     name ='FND_REQUEST' and line < 3

NAME          TYPE        TEXT
--------------- ----------- ----------------------------------------
FND_REQUEST     PACKAGE     package
FND_REQUEST AUTHID CURRENT_USER as
ND_REQUEST      PACKAGE     /* $Header:
AFCPREQS.pls 115.4 2000/02/29 11:51:08 $ */
FND_REQUEST     PACKAGE BODY package body
FND_REQUEST as
FND_REQUEST     PACKAGE BODY /* $Header:
AFCPREQB.pls 115.27 2001/07/28 09:35:16 $ */
```

FAQ

- How can you affect the look and feel of your Apps forms?

<u>appsweb.cfg</u>

lookAndFeel=Oracle (or generic)

colorScheme=blue or (teal, titanium, red, khaki, blue, olive, purple)

background=no , readonlyBackground=automatic

<u>Change System profile (after FND.D)</u>

Java Look and Feel=Oracle (or generic)

Java Color Scheme=blue (or teal, titanium, red, khaki, blue, olive, purple)

FAQ

- **Where can you find the source Forms modules for Apps?**

  Apps places all its source (FMB) files in the $AU_TOP/forms/<language> directory whereas the FMX files reside in their respective $PRODUCT_TOP/forms/<language> directory. On the other hand, all menu (MMB, MMX) and library (PLL and PLX) files are copied to the $AU_TOP/resource directory.

- **How can you find the version of Forms in Apps?**

  Once you are in an Apps form ...
  click on Help
  click on About Oracle Applications

FAQ

Oracle Applications Developer's Guide

Oracle Applications User Interface Standards for Forms=Based

CEMLI

- All concurrent program should register under XBOL application
- No database link is allowed, use flat file
- All customized object should create under BOLINF schema
- All program/files should be created under $XBOL_TOP
- Application User ID for oracle consultant should pre-fix with BOL- e.g.BOL-jsmith

CEMLI naming standard

Application Objects

Application User Accounts	BOL-DDDDD	D->Usr Acc
Flexfields	XXXX_DDDDDD	D ->Desc
Conc Programs	XXXX_DDDDDD	
Request Sets	XXXX_DDDDDD	
Responsibilities	XXXX_DDDDDD	
Menus	XXXX_DDDDDD	
Alerts	XXXX_DDDDDD	
Workflow	XXXX_DDDDDD	

Database Objects

Triggers, Index, Procedure, Sequence, Stored Procedure, Table, View

ORACLE

CEMLI Naming Convention

Operating System Files

Database Object Scripts, Database Trigger Script, Form , Reports, Form/Report Library, Grant Script, HTML file, Java files, SQL Loader Script etc