

Oracle Forms Development

Extending Applications the Right way

NCOAUG
Feb 28, 2005
Arlington Heights, IL
Susan Behn
www.solutionbeacon.com

Introduction

- Solution Beacon, LLC
 - ◆ Susan Behn

Outline

- Why Follow Standards?
- Assumptions
- Getting Started
- Property Classes
- Development Standards
- CUSTOM.pll
- Additional Resources
- Summary

Why Follow Standards?

- Faster development
- Consistent user interface
- Web compatibility
- Easier maintenance
- Compliance with American Disabilities Act
- Language and Currency Translation

Assumptions/Scope

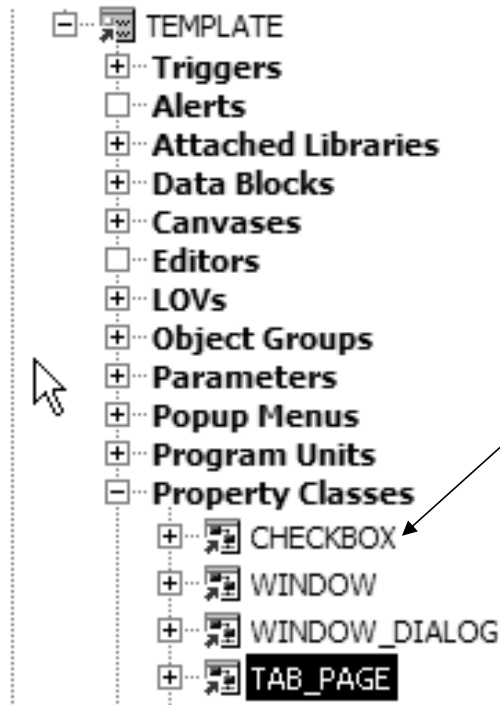
- Development environment is Applications Release 11i/Forms *6i*

- You are familiar with Forms Builder *6i*

*Oracle E-Business Suite application releases 11.x will continue to use *6i* (This includes 11.5.10)

Getting Started – TEMPLATE.fmb

- Download from \$AU_TOP/forms/US and rename
- Provides standardized sample objects and references
- Platform independent



Reference to object
in APPSTAND

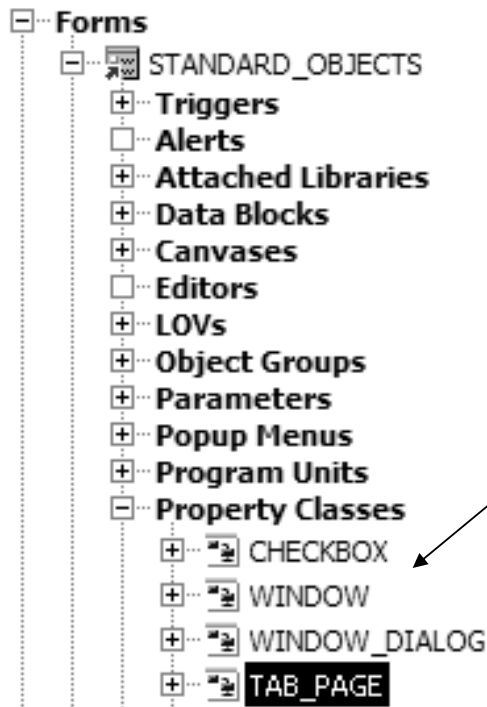
*Referenced Objects are
identified by the red
arrow in the lower left
corner

Getting Started – TEMPLATE.fmb

- Referenced objects (object groups)
 - ◆ LOVs, blocks, parameters, property classes
- Sample objects
 - ◆ Canvases, blocks, windows
- Form level triggers with required code
- Toolbar
- Menu
- Calendar
- Color palette
- Program unit APP_CUSTOM – standard behavior

Getting Started – APPSTAND.fmb

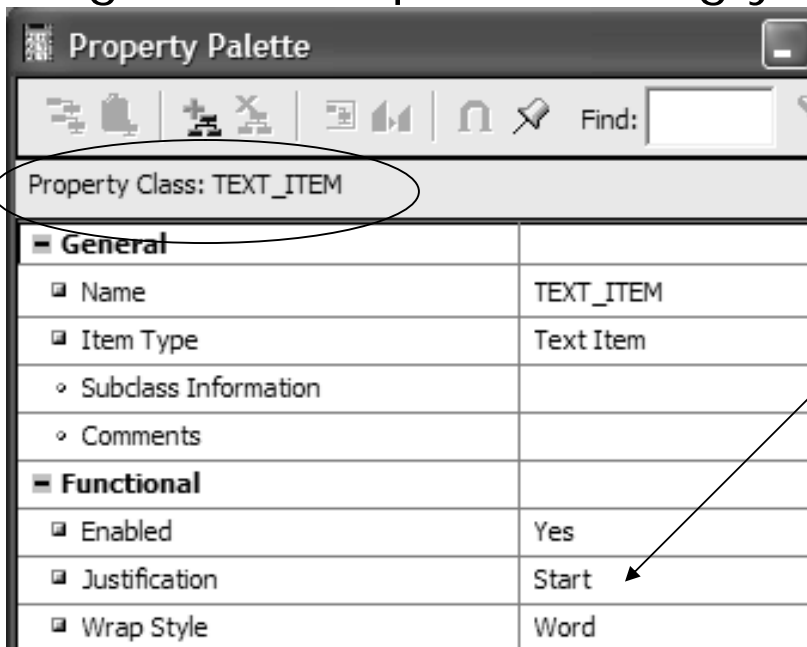
- Download from \$AU_TOP/forms/US
- Platform specific referenced objects
- References other libraries



Referenced
Object

Property Classes

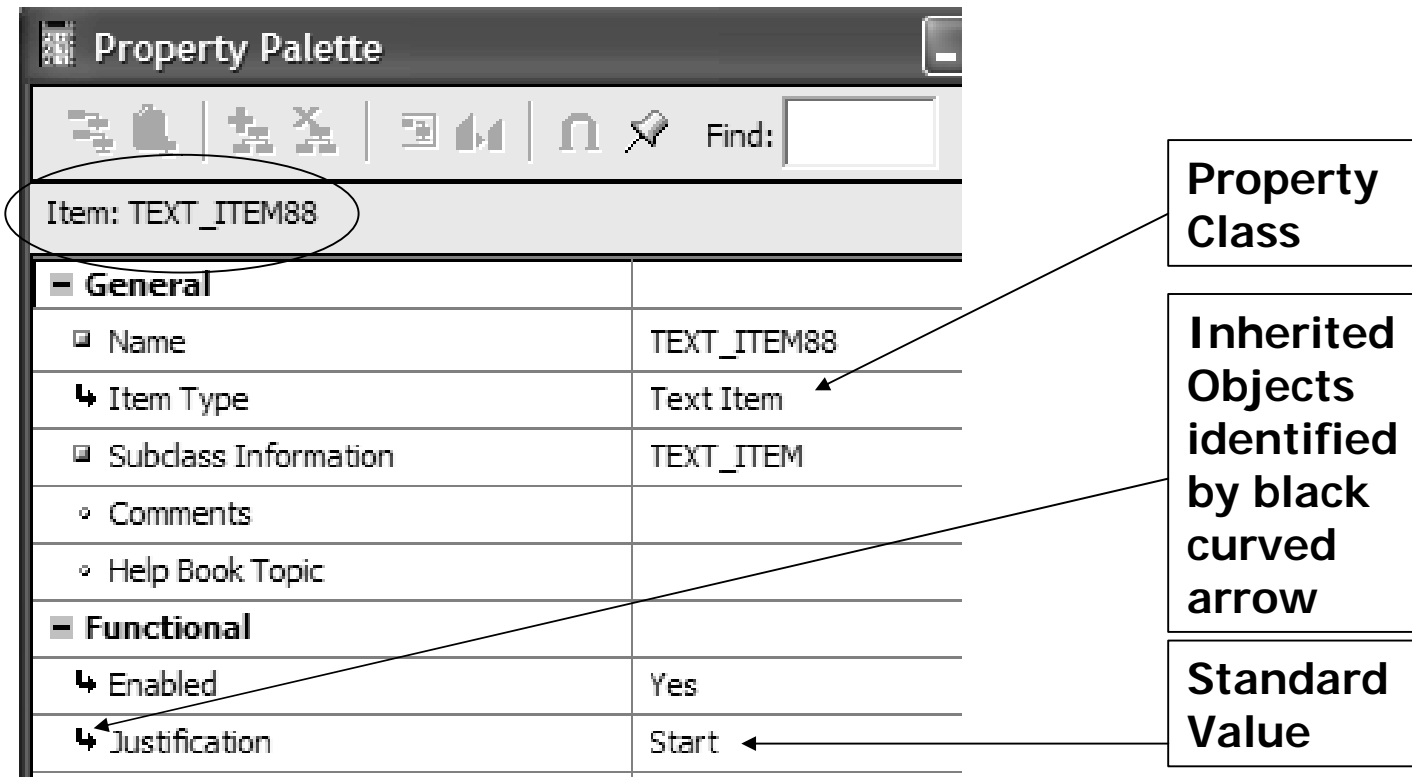
- Provide visual and behavioral attributes for objects
- Available for almost every object type
- Using property classes from these .fmb files is the single most important thing you can do to standardize!



Standard Value

Property Classes

- Do not override inherited property class attributes!



Property Palette

Item: TEXT_ITEM88

General	
Name	TEXT_ITEM88
Item Type	Text Item
Subclass Information	TEXT_ITEM
Comments	
Help Book Topic	
Functional	
Enabled	Yes
Justification	Start

Property Class

Inherited Objects identified by black curved arrow

Standard Value

Standards for Form Components

Module

Libraries

Triggers

Windows

Canvases

Data Blocks

Layout

Add'l Code

Module

Module

Libraries

Triggers

Windows

Canvases

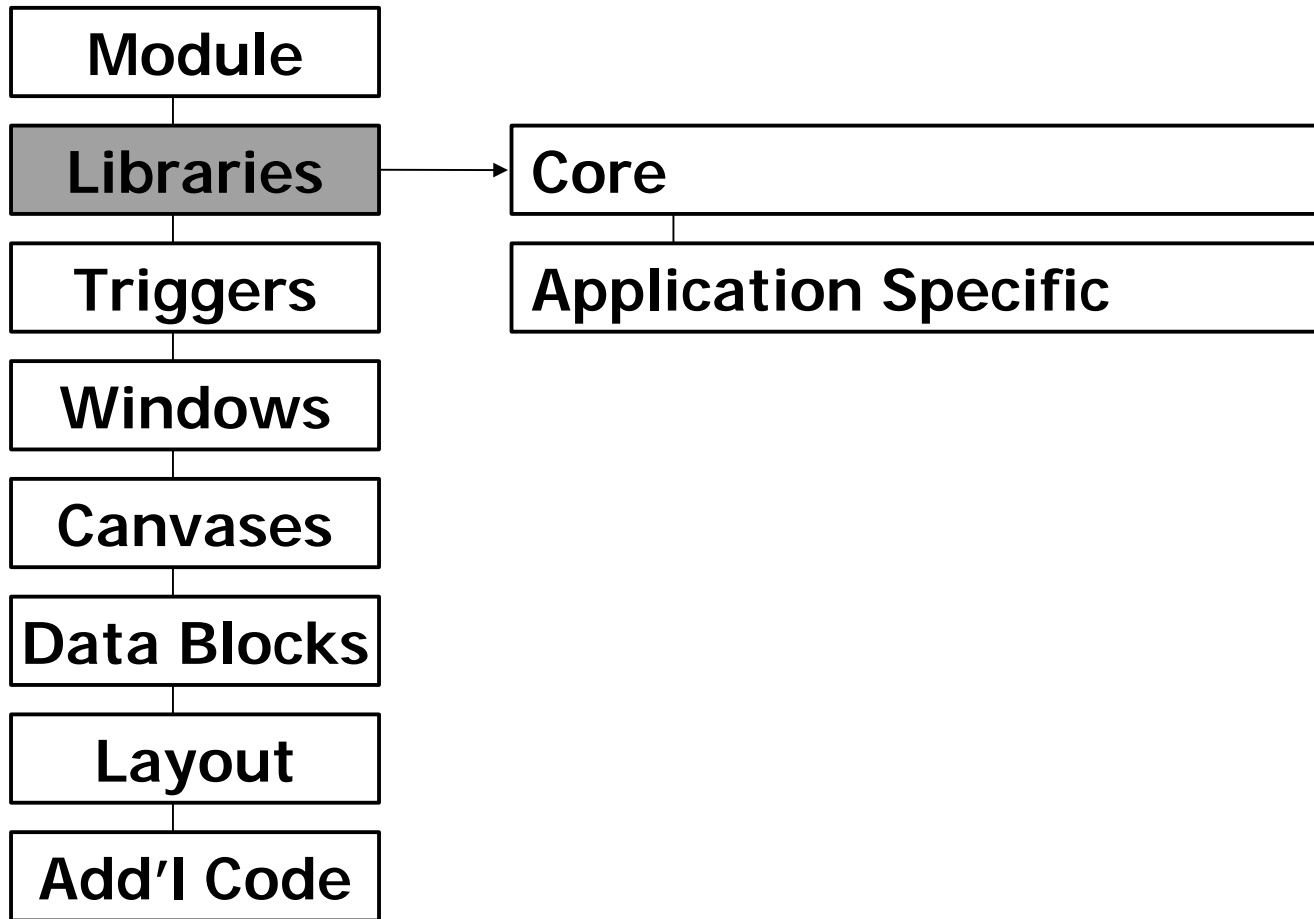
Data Blocks

Layout

Add'l Code

- Change the name of TEMPLATE.fmb
- Module name must match file name
- Property class = Module
- Set First Navigation Block

Libraries



Libraries – Core

- Download from \$AU_TOP/resource
- Set FORMS60_PATH on desktop to location of local libraries

- APPCORE.pll
 - ◆ VERT.pll
 - Application Standards, toolbar
 - Specific industry features
 - ◆ GLOBE.pll
 - Global or regional features
 - ◆ JA.pll
 - Asia/Pacific features
 - ◆ JE.pll
 - Europe/Middle East/Africa features
 - ◆ JL.pll
 - Latin America Code
 - ◆ CUSTOM.pll
 - Customizations of standard forms
 - FNDSQF.pll
 - Flexfields, Profiles, Currency,
 - Concurrent Request Submission,
 - Record History, Message Dictionary
 - APPDAYPK.pll
 - Calendar
 - APPFLDR.pll
 - Packages for Folder Blocks

Libraries – Application Specific

- Additional libraries attached to TEMPLATE.fmb
- All libraries are available in \$AU_TOP/resource

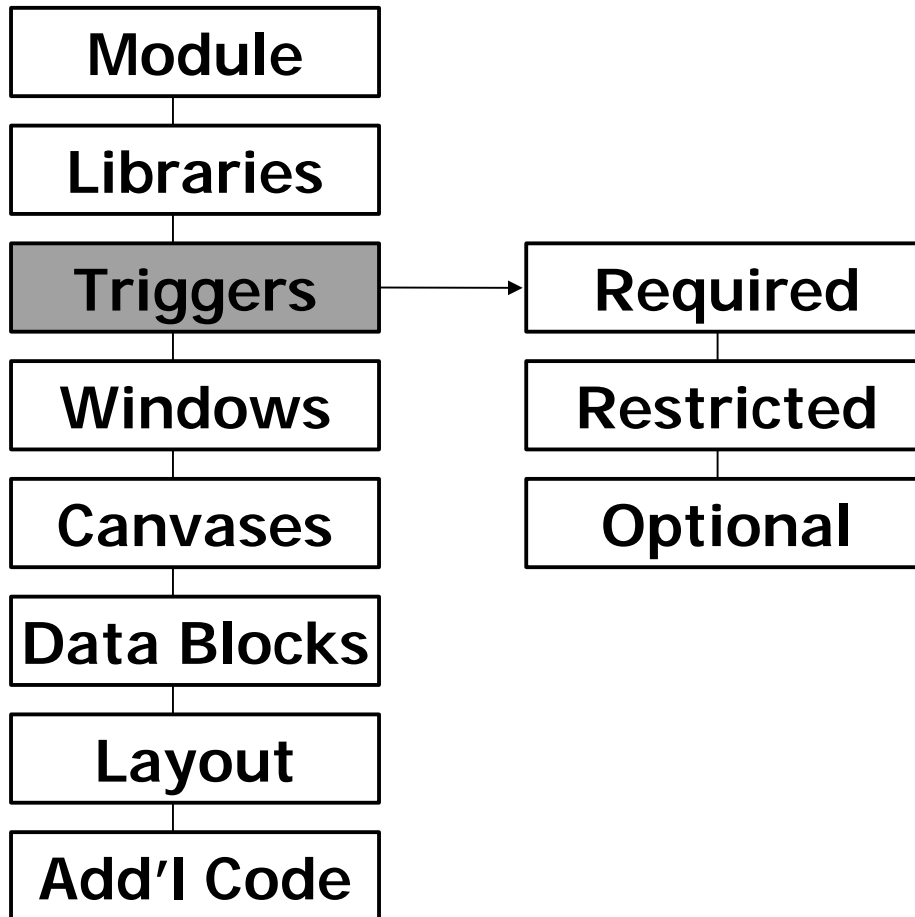
- | | |
|-----------------|--|
| ■ FV.pll | Federal Financials |
| ■ HRKPI.pll | Human Resources Knowledge Provider |
| ■ GHR.pll | Global Human Resources |
| ■ GMS.pll | Encumbrances |
| ■ IGILUTIL.pll | International Public Sector |
| ■ IGILUTIL2.pll | International Public Sector |
| ■ PSAC.pll | Public Sector Applications |
| ■ PQH_GEN.pll | Public Sector Human Resources |
| ■ PSA.pll | Public Sector Applications |
| ■ PSB.pll | Public Sector Budgeting |
| ■ OPM.pll | Process Manufacturing |
| ■ APPCORE2.pll | Application standards (for CUSTOM.pll) |

Libraries – Application Specific

- List of application specific libraries are not all inclusive
- Additional libraries are attached based on your implementation
 - Multiple Countries
 - Oracle Industry Applications
- Oracle does not support the use of APPFLDR, VERT, GLOBE, PSAC, PQH_GEN, GHR, JA, JE and JL for custom forms.

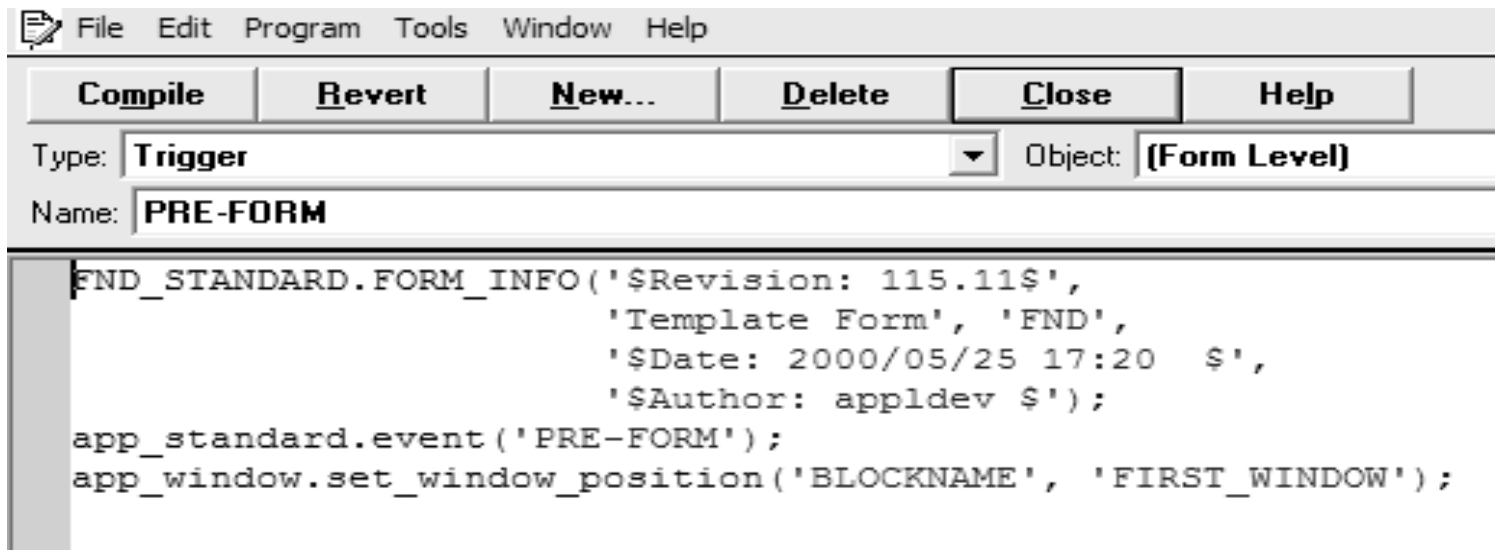
NEVER MODIFY ORACLE APPLICATIONS LIBRARIES OTHER THAN CUSTOM.pll

Triggers



Triggers – Required

- PRE-FORM must be modified
 - ◆ Revision, Date, Author
 - ◆ Template Form – change to your form name
 - ◆ BLOCKNAME – change to the first block
 - ◆ FIRST_WINDOW – change to the first window



The screenshot shows a window titled 'File Edit Program Tools Window Help'. Below the title bar are buttons for 'Compile', 'Revert', 'New...', 'Delete', 'Close', and 'Help'. The 'Type' is set to 'Trigger' and the 'Object' is '(Form Level)'. The 'Name' is 'PRE-FORM'. The main text area contains the following PL/SQL code:

```
FND_STANDARD.FORM_INFO('$Revision: 115.11$',  
                        'Template Form', 'FND',  
                        '$Date: 2000/05/25 17:20 $',  
                        '$Author: appldev $');  
app_standard.event('PRE-FORM');  
app_window.set_window_position('BLOCKNAME', 'FIRST_WINDOW');
```

Optionally Modify at Form Level

Do NOT delete the APP_STANDARD.EVENT call

- KEY-CLRFRM – add code after standard call
- POST-FORM – add code before standard call
- QUERY-FIND - add code after standard call
 - ◆ It's preferable to create a block-level trigger to override the form level QUERY_FIND trigger
- ACCEPT - add code after standard call

Don't Touch These Anywhere!

- STANDARD_ATTACHMENTS
- ZOOM
- FOLDER_ACTION
- KEY-HELP
- KEY-EXIT
- KEY-EDIT
- KEY-COMMIT
- WHEN-WINDOW-CLOSED
- CLOSE_WINDOW

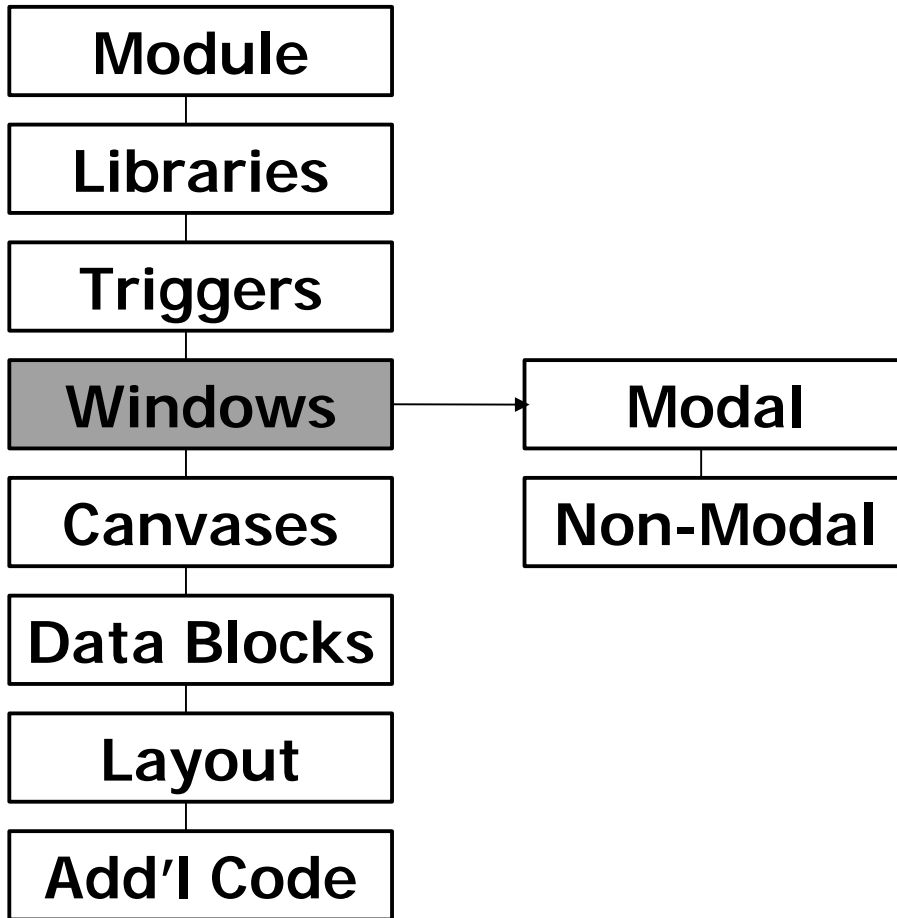
Optional Block or Item Level Triggers

- WHEN-NEW-RECORD-INSTANCE
- WHEN-NEW-BLOCK-INSTANCE
- WHEN-NEW-ITEM-INSTANCE
- POST-QUERY
- KEY-DUPREC
- KEY-MENU
- KEY-LISTVAL
- QUERY-FIND
- ACCEPT
- ON-ERROR

Triggers – Execution Hierarchy

- Execution Hierarchy – Default is OVERRIDE
 - ◆ NEW triggers – set to BEFORE
 - ◆ POST triggers – set to AFTER
 - ◆ KEY triggers – set to OVERRIDE
 - ◆ QUERY-FIND triggers – set to OVERRIDE

Window



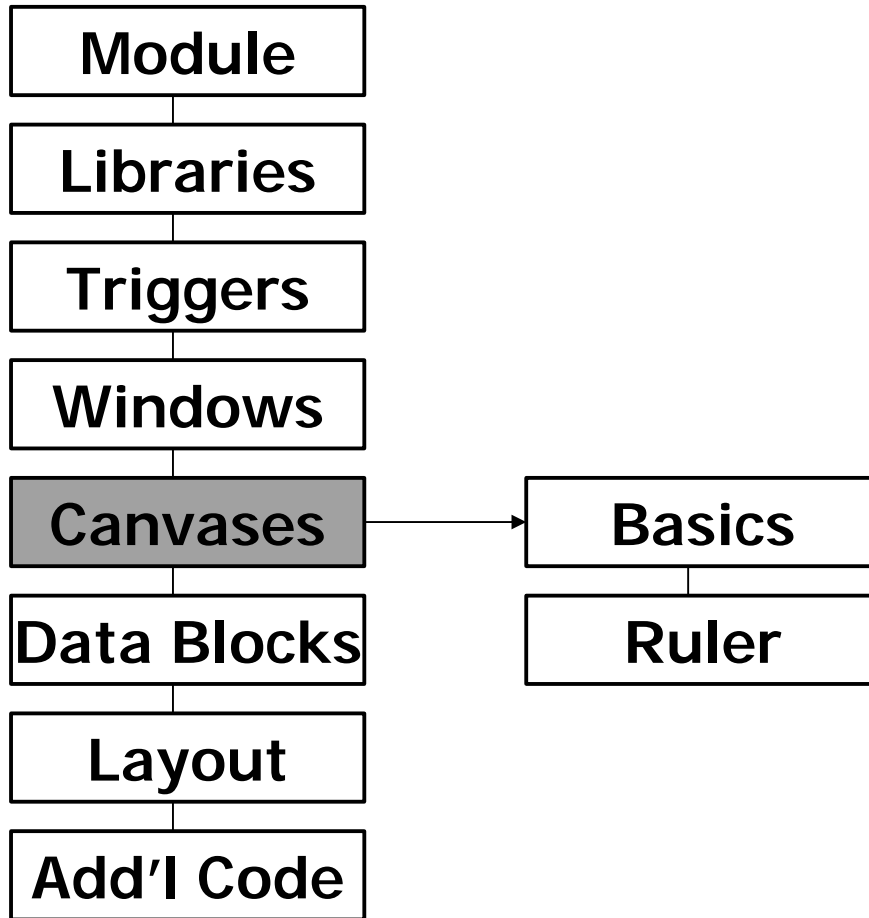
Modal Windows

- Property class = WINDOW_DIALOG
- Center window using
APP_WINDOW.SET_WINDOW_POSITION
- Explicitly write code to close the window
- Code OK and Cancel buttons in advance
- Use dialog blocks with property class =
BLOCK_DIALOG
- Disable key triggers

Non-Modal Windows

- Property class = WINDOW
- Each window should contain one logical entity
- Do not attach scroll bars to windows – attach to blocks
- Size between 2"x2" and 7.8"w x 5.0"h
- Size only as large as necessary
- No text on top and bottom lines
- Blank left and right edge character cell except for boundary lines
- Unique title for each window in a form
- Don't use only singular/plural forms on window titles
- Don't change the word order on window titles
- Don't pad window titles with spaces

Canvases



Canvas – Basics

- Property class = CANVAS, TAB_CANVAS or CANVAS_STACKED
- Each window contains one content canvas
- Tab or stacked canvases may be placed in front of a content canvas
- Utilize tabs to avoid scrolling

Canvas – Ruler

■ Ruler Settings

Ruler Settings

Units

Inches Points

Centimeters Character Cells

Character Cell Size (points)

Horizontal: 7.2

Vertical: 18

Grid Spacing

0.25 0.5 1 2 5 Other:

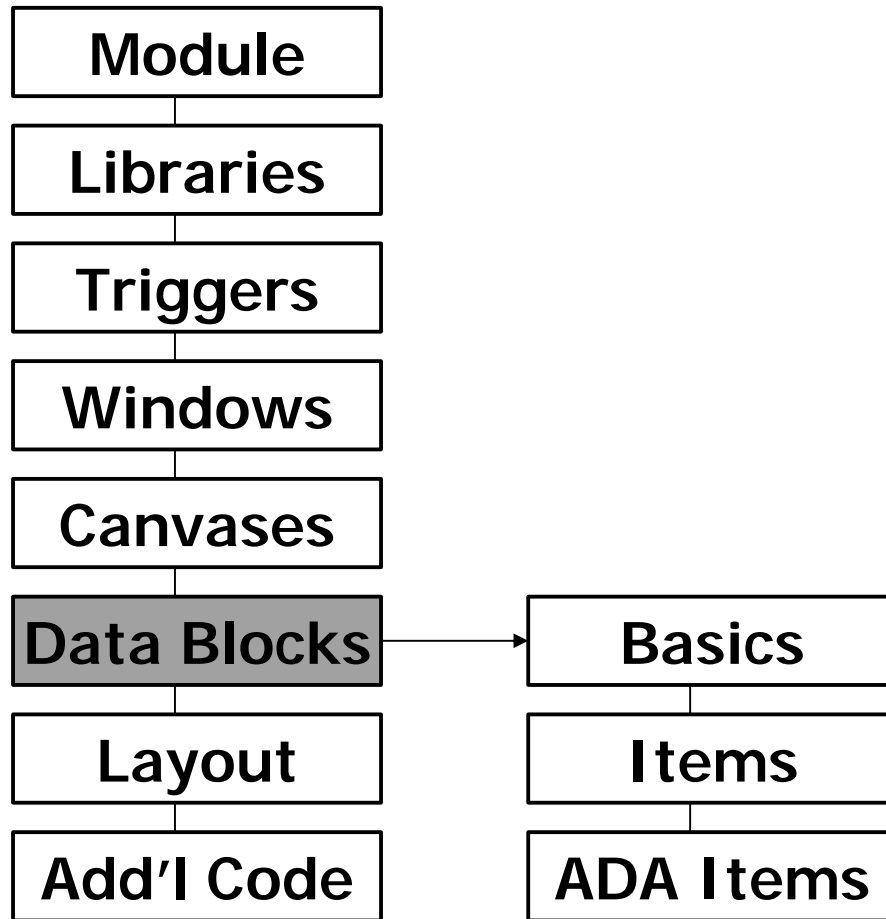
Number of Snap Points Per Grid Spacing

1 2 4 5 8 Other:

OK Cancel Help

- Character cell height = .25
- Character cell width = .1
- Turn snap to grid on in the View drop down

Data Blocks



Data Blocks – Basics

- All blocks for a logical entity should be in one window
- Exceptions
 - ◆ Information not perceived as attributes of the logical entity
 - ◆ Fields not frequently needed
- Use tabs or alternating regions to avoid horizontal scrolling
- For multi-record blocks always add a scroll bar and current record indicator

Data Blocks – Basics

- Base complex data blocks on views
 - ◆ Network traffic is minimized because foreign keys are denormalized on the server
 - ◆ POST-QUERY code for non-database fields eliminated
 - ◆ PRE-QUERY code for non-database fields eliminated
 - ◆ Allows for multi-org and multi-language
- Base simple blocks on tables
- Navigation style
 - ◆ One block: Current Record
 - ◆ More blocks: Next Block

Data Block – Items

- Enable the calendar for every date field
 - ◆ Use ENABLE_LIST_LAMP LOV
 - ◆ Call calendar_show from KEY-LISTVAL trigger
- Control valid dates
- Check boxes always have a default value
- Maximum size of items = database column size
- Validate items immediately when possible
- LOVs
 - ◆ LOVs should only show valid rows
 - ◆ Validate from List = Yes for items with LOVs
 - ◆ Base LOVs on views to allow sharing
 - ◆ The title in the first item of the LOV should match the prompt of the field calling the LOV

Data Block – Items

■ Buttons

- ◆ Place buttons in the block in which they appear to sit
- ◆ Provide an access key
- ◆ For multi-row blocks, set Keyboard Navigable to No
- ◆ For single row blocks, set Keyboard Navigable to Yes
- ◆ In the WHEN-BUTTON-PRESSED trigger, set “Fire in Enter Query mode” to No

Data Block – Items

Compliance with American Disabilities Act

- *All actions must be executable using the keyboard*
 - ◆ Specify access keys for all non-navigable items using & in the label property for:
 - *Buttons
 - *Radio buttons
 - *Check boxes
 - *Tab pages
 - ◆ Navigation is left to right, top to bottom
 - ◆ Cycle through all fields on multi-record blocks (eliminates need for scroll bar)
- *Timed functions are prohibited*
- *The user must be able to control color – Use defined property classes*

Data Block – Items

Compliance with American Disabilities Act

- *Sufficient information must be available to assisted technology (i.e. screen readers)*

Examples of items to specifically address

Flexfield high/low ranges

– Effective Dates –

From	To
<input type="text"/>	<input type="text"/>

Abbreviated prompts

Num	Org
<input type="text" value="1"/>	<input type="text" value="V1"/>

Non-unique prompts, Region titles

Bank	Branch
Name <input type="text"/>	Name <input type="text"/>
Number <input type="text"/>	Number <input type="text"/>

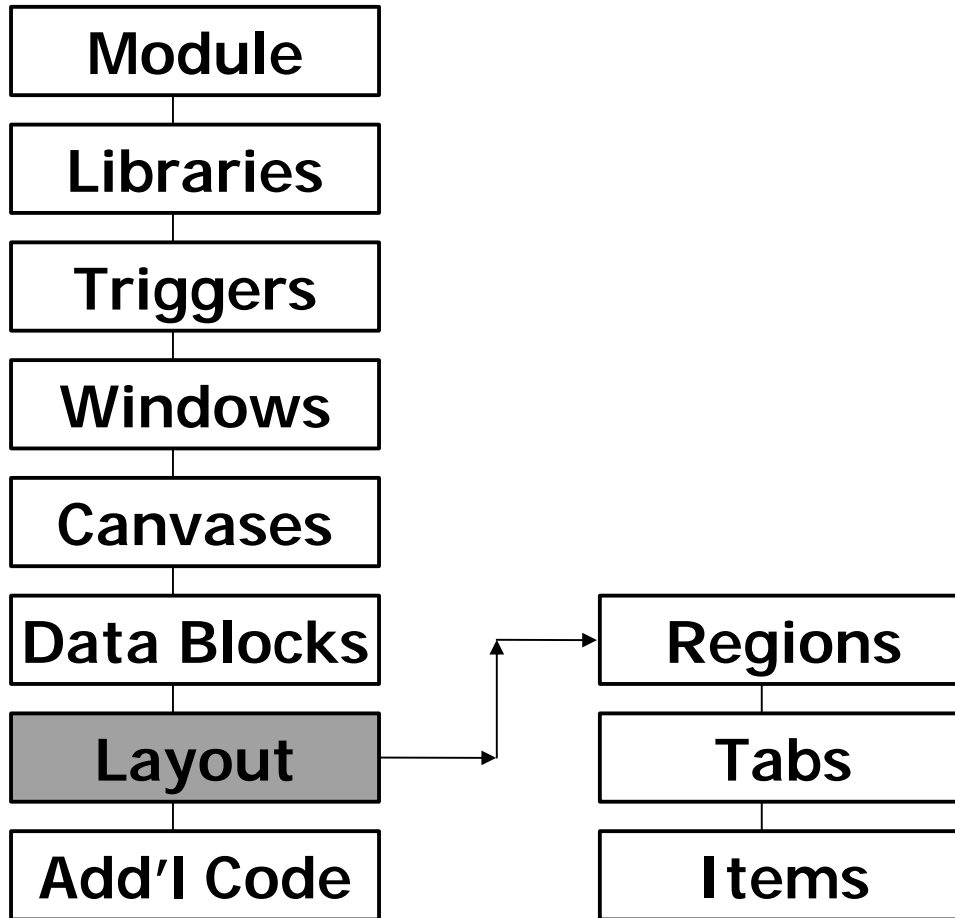
Data Block – Items

Compliance with American Disabilities Act

- How is this accomplished?
 - ◆ Enter hint text for items
 - ◆ Optionally enter tool tip help

- Prompts are read in the following order:
 - ◆ Hint Text
 - ◆ Prompt
 - ◆ Label
 - ◆ Tool tip help

Layout



Layout – Items

■ Items

- ◆ Leftmost button in the window should be the default unless it is a help button
- ◆ Allow 30% additional text space for language translation
- ◆ Set alignment to start

■ Tabbed regions

- ◆ One character space between fixed and alternating regions
- ◆ Place coordination check box outside the tabbed region

■ Navigation: Left to right, top to bottom

Layout – Regions

The screenshot shows the Oracle Invoices application window. The title bar reads "Invoices (Vision Operations: USD)". The menu bar includes "File", "Edit", "View", "Folder", "Tools", "Window", and "Help". The toolbar contains various icons for file operations. The main area features a table with the following data:

Type	Supplier	Supplier Num	Site	Invoice Date	Invoice Num	Invoice Curr	Invoice Amount
Standard	Advanced M	1013	SANTA C	11-FEB-1998	10112	USD	2,320.70
Standard	Advanced M	1013	SANTA C	02-DEC-2001	1013-1-Week	USD	107,800.00
Standard	Advanced M	1013	SANTA C	09-DEC-2001	1013-1-Week	USD	108,069.50
Standard	Advanced M	1013	SANTA C	16-DEC-2001	1013-1-Week	USD	108,339.00
Standard	Advanced M	1013	SANTA C	23-DEC-2001	1013-1-Week	USD	108,610.50
Standard	Advanced M	1013	SANTA C	13-JAN-2002	1013-2-Week	USD	107,600.00

Below the table, there are summary fields: "Amount Paid" (USD 2,320.70), "Holds" (0), "Status" (Validated), "Approval" (Not Required), "Distribution Total" (2,320.70), and "Accounted" (Yes). At the bottom, there are buttons for "Actions... 1", "Holds", "Payments", "Match", "Purchase Order" (dropdown), "Scheduled Payments", "Overview", and "Distributions".

Stacked Alignment

Coordination Check Box

Scroll Bar

Current Record Indicator

Region Frame



Layout – Tabs

- Tab Pages

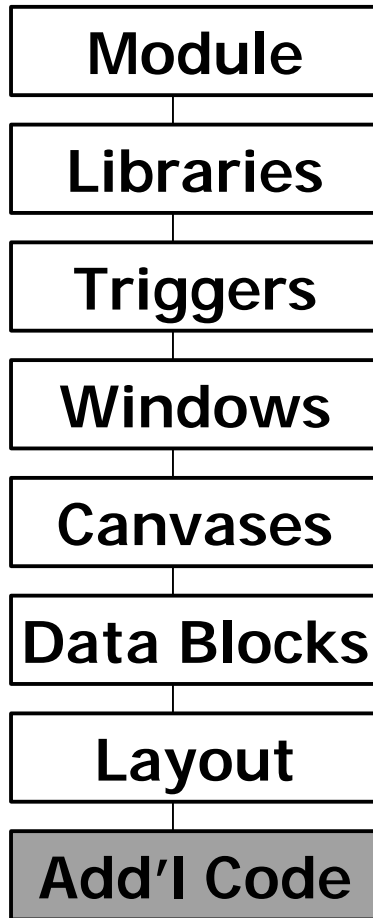
The screenshot shows the Oracle Shipment Management interface. At the top, there are tabs for 'Shipments', 'More', and 'Status'. Below the tabs is a table with the following columns: Num, Org, Ship-To, UOM, Quantity, Promised Date, Need-By, and Original Promise. The first row of the table contains the following data: Num: 1, Org: M1, Ship-To: V1- New York, UOM: Week, Quantity: 1, Promised Date: (empty), Need-By: (empty), Original Promise: (empty). Below the table, there are input fields for 'Line Num' (1), 'Item' (z7445), and a description 'Cleaning and Maintenance - Weekly'. At the bottom, there are two buttons: 'Receiving Controls' and 'Distributions'. A callout box labeled '1 character cell' points to the 'Line Num' field.

Num	Org	Ship-To	UOM	Quantity	Promised Date	Need-By	Original Promise
1	M1	V1- New York	Week	1			

One Line

1 character cell

Additional Code



Use program units!

Additional Code – program units

- Do not base code on a translated field – use the hidden value
- Use “who” fields and update them!
- If more than one line of code is required, call a program unit, otherwise, code directly in the trigger

CUSTOM.pll

- Used to provide customizations to Oracle Application forms that are protected from patches and upgrades
- Traditional implementation of customizations in CUSTOM.pll only allow one developer at a time to make modifications
- Size limitations can be an issue for implementations with many customizations
- CUSTOM.pll can be used with custom forms as long as they were developed per standards using TEMPLATE.fmb

CUSTOM.pll

■ Multi-Developer Methodology

Create a .pll library for the form you want to customize

Package Spec

```
PACKAGE XXXXXAPXVDMVX IS
  Procedure event(event_name VARCHAR2);
END;
```

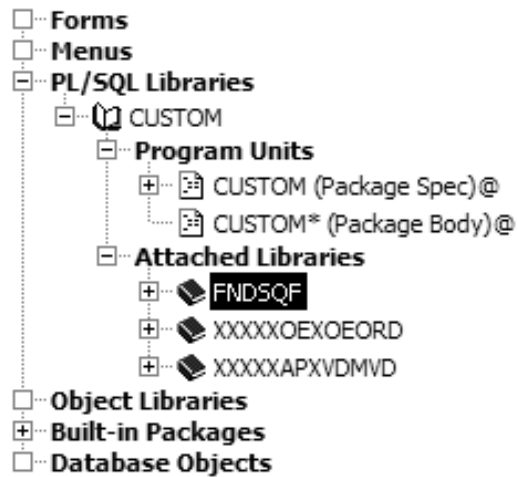
Package Body

```
PACKAGE BODY XXXXXAPXVDMVX
  IS
  PROCEDURE event (event_name VARCHAR2) IS
    BEGIN
      IF event_name = 'WHEN-NEW-FORM-INSTANCE' THEN
        SET_ITEM_PROPERTY('VENDOR_TYPE_DISP',REQUIRED,PROPERTY_TRUE);
      END IF;
    END event;
  END XXXXXAPXVDMVX;
```

CUSTOM.pll

■ Multi-Developer Methodology

Attach your .pll to CUSTOM.pll



Navigator view of
CUSTOM.pll

CUSTOM.pll

■ Multi-Developer Methodology

Add the call to your new .pll to CUSTOM.pll

Package Body of CUSTOM.pll

```
Form_name      varchar2(30) := name_in('system.current_form');  
Begin  
  If form_name = 'XXXXXAPXVDMVD' THEN  
    xxxxxapxvdmvd.event(event_name);  
  Elif form_name = 'XXXXXOEXOEOORD' THEN  
    xxxxxoexooord.event(event_name);  
  end if;  
end event;
```

Other Sources of Information

- *Oracle Applications User Interface Standards for Forms-Based Products*
- *Oracle Applications Developer's Guide*
- *Oracle Applications System Administrator's Guide*
- *Oracle Applications User Guide*
- *MetaLink*

Patches – Client

- *Patch set 15 is most current*
- *Patch set 13 – absolute minimum!*

Summary

- Start with TEMPLATE.fmb
- Always apply property classes
- Set trigger execution hierarchy correctly
- Follow layout standards
- Use program units to keep code modular
- Make it look good!

Thank you!

If you have any questions or comments please contact:

Susan Behn sbehn@solutionbeacon.com

For free R11/Tools please visit our website at:

www.solutionbeacon.com