**Object Sharing in Forms 10G**

When you share and reuse objects and code, you decrease maintenance, increase modularity, and improve the performance of your applications. When you are developing applications, you should share and reuse objects and code wherever possible.

There are many advantages of sharing and reusing objects and code.

* You can develop applications much more effectively and efficiently if you are not trying to “start over” each time you write a piece of code. By sharing and reusing frequently used objects and code, you can cut down development time and increase productivity.
* You can decrease maintenance time by creating applications that use or call the same object or piece of code several times.
* You can maintain standards by reusing objects and code.

To maximize reuse, the developer should apply the following guidelines in the order shown:

* Accept default properties as much as possible
* Use SmartClasses to describe an object
* Use sets of visual attributes

***Property Class***

**A property class is a named object that contains a list of properties and their settings/values**.

You can use property classes to increase productivity.

 Here are some of the uses of a property class:

* It increases productivity by setting standard or frequently used values for common properties and associates them with several form builder objects.
* It defines standard properties not just for one particular object but also for several at a time. This results in increased productivity, because it eliminates the time spent on setting identical properties for several objects.
* It improves network performance by increasing the efficiency of message diffing.

**Message Diffing :**  Once an item is defined, the meta-data about the next item includes only those properties that differ from the previous item. This is referred to as message diffing.

Steps for Property Class

When you create a property class, you have all the properties from every Forms Builder object available. You choose the properties and their values to include in the property class. You can create a property class in two ways:

* Using the Create button in the Object Navigator
* Using the Create Property Class button in the property-palette

**Steps to create a property class from the Object Navigator:**

Create a new form module. Save it as F34PropClass. Create a block manually and name the block as CTRLBLOCK. Add two text items to the block and name them as TEXT1 and TEXT2. Create a content canvas and name it as CONTCANVAS.

In the Object Navigator, click on Property Classes and click the create(+) button. Name the Property Class as :MYPROPCLASS.

In the property palette of MYPROPCLASS, click the (+) button to add the property “Canvas” and set it’s value to “CONTCANVAS”. Similarly, add the property “Data Type” and set it’s value to “Number”. Similarly, Maximum Length=20, Initial Value=0, Required=Yes, Database Item=No, Width=50, Height=20, Font Name=Arial, Font Size=12, Foreground Color= r88g0b0, Background color= r0g100b88.

Select TEXT1, and for the property “Subclass Information”, select the radio-button “Property Class”, and take the “Property Class Name” as MYPROPCLASS. Now, notice the above properties(properties in the property class MYPROPCLASS) for the TEXT1 item.

**Steps to create a property class from the Property Palette:**

Select the TEXT2 and set the following properties :

Canvas = CONTCANVAS, Data Type=date, Maximum Length=10, Initial Value=28-Feb-06, Required=Yes, Database Item=No, Width=40, Height=15, Font Name=Courier, Font Size=14, Foreground Color= r0g0b50, Background color= r100g50b100.

Now, in the property palette of TEXT2, select all the above properties, by keeping the CTRL key pressed and selecting all the above properties, and click on the “Property Class” button at the top of the property palette of TEXT2. A new property class gets created. Select it in the ON, and name it as YOPROPCLASS.

Create a text item in the CTRL BLOCK, name it as TEXT3.

Select TEXT3, and for the property “Subclass Information”, select the radio-button “Property Class”, and take the “Property Class Name” as YOPROPCLASS. Now, notice the above properties(properties in the property class YOPROPCLASS) for the TEXT3 item.

Notice, how TEXT1 and TEXT3 have small red arrows in the Object Navigator.

All the inherited properties are shown with a black arrow in the property palettes of TEXT1 and TEXT3. If you change any inherited property, say you change initial value=06-Mar-01 for TEXT3, that particular property is shown with a red cross at the tip of the black arrow.

**What is an inherited property?**

An *inherited property* is one that takes its value from the property class that you associated with the object. An inherited property is displayed with a black arrow to the left of the property name.

**What is a variant property?**

A *variant property* is one that has a modified value even though it is inherited from the property class associated with the object. You can override the setting of any inherited property to make that property variant. Variant properties are displayed with a red cross over the black arrow.

* To convert an inherited property to a variant property, simply enter a new value over the inherited one.
* To convert a variant property to an inherited property, click the “Inherit” icon at the top of the Property Palette.

***Copying and Subclassing Objects and Code***

With subclassing you can make an exact copy, and then alter the properties of some objects, if desired. If you change the parent class, the changes also apply to the properties of the subclassed object that you have not altered. However, any properties that you override remain overridden. This provides a powerful object inheritance model.

SubClassing an Object

Select the TEXT1. Hold down the CTRL key, keep the left mouse button pressed and drag it. The moment you leave the left mouse button, it will ask you whether you want to “Subclass”, “Copy” or “Cancel”. Select the “Subclass” option. Name this new item as TEXT4. You will notice that the “Subclass Information” property= TEXT1 for TEXT4.

For TEXT4, set/change the following properties :

Maximum Length=22

Initial Value=1

For these properties, you will notice the black arrow with red cross.

Now, for TEXT1, set/change the following properties :

Width=55

Height=25

Initial Value=100

Now, notice the above 3 properties for TEXT4. You will notice that

Width=55

Height=25

Initial Value=1

Thus, for TEXT4, Height and Width have automatically changed to 55 and 25(as per the changes made to TEXT1), however, Initial value continues to remain 1(despite of Initial value being set to 100 for TEXT1)

Copying an Object

Select the TEXT3. Hold down the CTRL key, keep the left mouse button pressed and drag it. The moment you leave the left mouse button, it will ask you whether you want to “Subclass”, “Copy” or “Cancel”. Select the “Copy” option. Name this new item as TEXT5. You will notice that the “Subclass Information” property is not TEXT3(but it is blank) for TEXT5.

Now, for TEXT3, set/change the following properties :

Maximum length=12

Now, go to TEXT5, the Maximum length is still 10.

You can copy or subclass objects:

* Within a single module by selecting the object in the Object Navigator, pressing [Ctrl], and dragging it to create the new object(as we did above)
* Between modules, by dragging objects between the modules in the Object Navigator

You may try this by creating a new form module, save it as F35PROPCLASS. In it, create a data block manually and name it as CTRLBLOCKnew. Now, simply drag any item, say TEXT3 from the module F34PROPCLASS into the CTRLBLOCKnew of F35PROPCLASS. And accordingly decide whether you want to Subclass or Copy

When you subclass a data block, you can:

* Change the structure of the parent, automatically propagating the changes to the child
* Add or change **properties** to the child to override the inheritance

When you subclass a data block you cannot:

* Delete **items** from the child
* Change the order of **items** in the child
* Add **items** to the child unless you add them to the end. That is, you can add items to the end of the list of items

**Copy or Subclass?**

When you copy an object, a separate, unique version of that object is created in the target module. Any objects owned by the copied object also get copied. Changes made to a copied object in the source module do not affect the copied object in the target module.

With subclassing, you can make an exact copy, and then alter the properties of some objects. If you change the parent class, the changes also apply to the properties of the subclassed object that you have not altered.

***Object Groups***

An *object group* is a logical container for a set of Forms Builder objects. **The smallest unit accepted in an object group is a block**. You need to define an object group when you want to package related objects for copying or subclassing in another module or if you want to bundle numerous objects into higher-level building blocks that you can use again in another application. You can package the various objects in an object group and copy or subclass the entire bundle in one operation.

**Steps to create an object group**

In F34PROPCLASS, in the ON, click on Object Groups, click on the create(+) button, name it as “MY\_OBJECT\_GROUP”. Select and drag the block CTRLBLOCK from the module F34PROPCLASS into the Object Group “MY\_OBJECT\_GROUP” just created.

Create a new form module and save it as F36PROPCLASS. From the F34PROPCLASS, select and drag the Object Group “MY\_OBJECT\_GROUP” into the module F36PROPCLASS, and then “Subclass” or “Copy” as per requirement. As of now, take the option “Subclass”. The CTRLBLOCK in the module F36PROPCLASS will have the red arrow, and its “Subclass Information” property will have the value CTRLBLOCK. Also each item in the CTRLBLOCK in F36PROPCLASS module will have the red arrow. You wont be able to delete the CTRLBLOCK or any of it’s items from the F36PROPCLASS module. But you can change any of the property values as required.

* Including a block in an object group also includes its items, the item-level triggers, the block-level triggers and the relations. You cannot use any of these objects(items, the item-level triggers, the block-level triggers and the relations) in an object group without the block.
* Deleting an object from a module automatically deletes the object from the object group.
* Deleting an object group from a module does not delete the objects it contains from the module.

***D) Object Libraries***

*Object libraries* are convenient containers of objects for reuse. They simplify reuse in complex environments, and they support corporate, project, and personal standards, as they are protected against change in the library. An object library can contain simple objects, property classes, object groups, and program units.

**Why object libraries instead of object groups?**

* Object libraries are external to the form, so are easily shared among form modules.
* Object libraries can contain individual items whereas the smallest unit accepted in an object group is a block.
* Object libraries accept PL/SQL program units.
* If you change an object in an object library, all forms that contain the subclassed object reflect the change. **Note:** You cannot modify objects inside the object library itself. To make changes, drag the object into a form, change it, and drag it back to the object library.

Object libraries appear in the Navigator if they are open. You can create, open, and close object libraries like other modules. Forms Builder automatically opens all object libraries that were open when you last closed Forms Builder.

Steps to show Object Library

In the F34PROPCLASS, in the CTRLBLOCK, create a new text item TEXT6, and set the following properties :

Foreground Color :r88g0b0

Background color : r0g100b0

Font Name : Arial

Font Style : Italic

Then, click on the Object Libraries module, and click on create(+), save it as MYOBJLIB. Under MYOBJLIB, click on Library Tabs, and click create(+) to create a new library tab. Then double click on the newly created tab of MYOBJLIB. Now, drag TEXT6 and MYPROPCLASS from F34PROPCLASS into the only Library tab of MYOBJLIB. Save the MYOBJLIB object library module.

Now, create a new form module and thru the data-block and layout wizard take the EMP table from scott schema in the form style with number of records displayed as 1. Save it as “EMP.fmb”. Select the ENAME text box, and go to it’s property palette. For the property “Subclass Information”, select the “Object” radio button. Under Module, take the option “MYOBJLIB”(ensure that the Object Library MYOBJLIB is open), and Object Name as TEXT6. You will notice that the Ename text box’s Foreground Color, Background color, Font Name, Font Style properties would be changed and have a red arrow in the ON and black arrows for the appropriate properties in the property palette also its Subclass Information property will be set as TEXT6.

**SmartClass**

A SmartClass is a special member of an Object Library. It can be used to easily subclass existing objects in a form using the SmartClass option from the right mouse button popup menu.

To use Object Library members which are not SmartClasses, you have to use the Subclass Information dialog available in the Property Palette of the form object that you are modifying(as we did in the above example)

If you frequently use certain objects as standards, such as standard buttons, date items, and alerts, you can mark them as SmartClasses by selecting each object in the object library and choosing Edit > SmartClass. You can mark many different objects, spread across multiple object libraries, as SmartClasses.

To apply a SmartClass to a Forms object, perform the following steps:

1. Right-click an object in the Layout Editor or Navigator.

2. From the pop-up menu, select SmartClasses. The SmartClasses pop-up menu lists all the SmartClasses from all open object libraries that have the same type as the object

3. Select a class for the object; it then becomes the parent class of the object. You can see its details in the Subclass Information dialog box in the object’s Property Palette, just like any other subclassed object.

Steps to create smartclass items from Object Library

In the Object library “MYOBJLIB” select TEXT6 item and click on the Menu EDIT, and select “SmartClass”. You will see a green tick mark for TEXT6 item, indicating that is an object of the MYOBJLIB which is available for smart-classing option.

In the EMP form module, right click on the JOB item and from the popup menu and take the option “SmartClasses”, and select the option TEXT6.