

An example of View Object Extension in conjunction with Controller class extension

This is a real life implementation example of OA Framework Extension on Release12.

In this extension, you will see how to

1. Extend a View Object to add a new Column
2. How to extend a controller class
3. How to apply substitution of View Object
4. How to export personalization and deploy the same

Given that this was a real life solution implemented for a client of mine, I am unable to make the business requirement a public knowledge. For the very same principles, I am not detailing the source code for the controller extension.

But yet this document will give you a good idea of how extension of BC4J, extension of Controller and Personalizations work in tandem.

Before you begin :-

1. Set environment variable JDEV_USER_HOME to C:\anilpassi\JDev_for_R12\jdevhome\jdev
2. copy the vis.dbc file from server to pc. (eg. Jdevbin\oaext\dbc_files\secure)
3. create database connection
4. open the workspace for tutorial and rebuild its projects, so that myclasses folder gets created

Solution Working steps

1. Create a button and a checkbox using personalization
2. Extend view object to add email and a column to support the new checkbox
3. Create the substitution in jDev and upload the jpx substitution to the Database
4. extend the controller, which does the following

<http://oracle.anilpassi.com>

Brief Steps - OA Fwk R12 Extension Example

processRequest

 Make the new checkbox fire an action

 Put a dummy javascript function

 Attach the dummy javascript function to the Button

processFormRequest

 Check if the checkbox event has fired

 If Yes

 Then loop through the ViewObject, for records with Checkbox equals Yes

 Construct the Email String

 Replace the definition of the javascript Function

5. Add the extended controller to the region using personalization

<http://oracle.anilpassi.com>

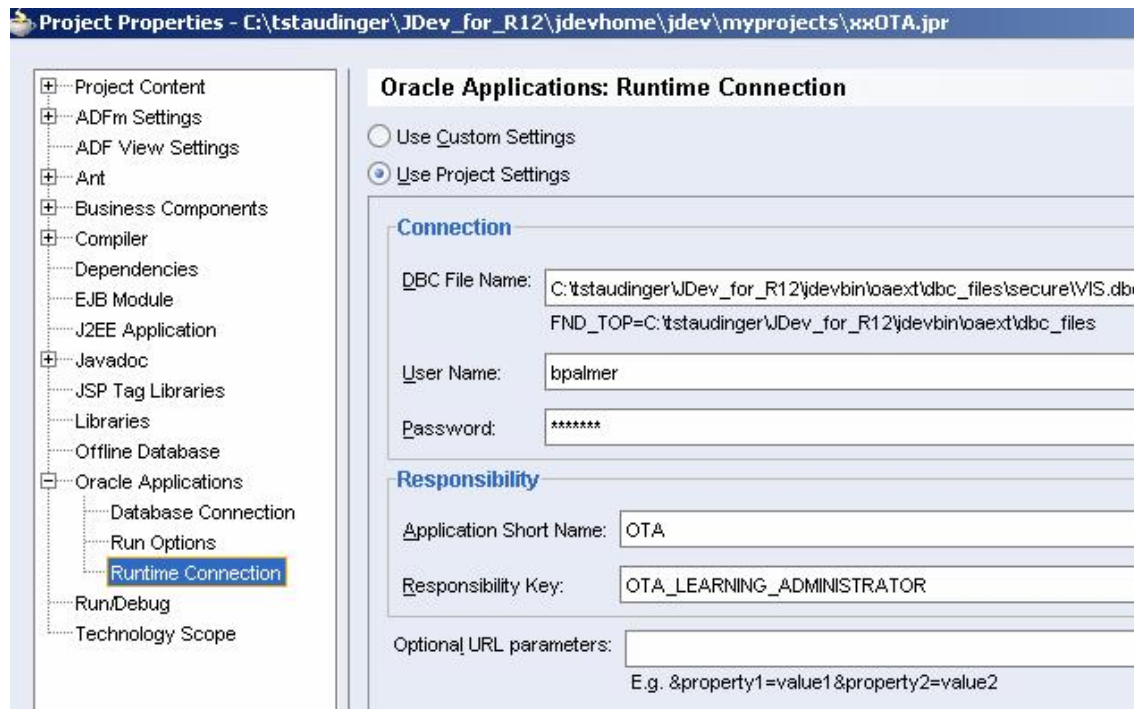
Brief Steps - OA Fwk R12 Extension Example

Deployment Steps

-
1. Deploy entire tree from myclasses/xxrzb to \$JAVA_TOP
Also copy the source code to \$JAVA_TOP
 2. Next, copy the jpx file and the upload script to \$xxrzb_top
 3. Upload the substitution to the DB using the jpx-upload script
 4. Add the Personalization for the controller

Setup steps for working on jDeveloper Extension

1. Create a new OA Workspace
2. Create new OA Project. In the jDeveloper project property, assign the DBC File, which was FTPed from server. Ensure that the username, password, responsibility application and responsibility Key combination is correct.



3. FTP the files from JAVA_TOP to myclasses. Make sure that dependent classes are copied as well.
4. Copy the files from myclasses to myprojects. Remove the .class files from the myprojects. Here, you need to copy just those files which you wish to extend.
5. FTP the xml files from \$PRODUCT_TOP/mds to myprojects



6. Change the Library path for jDeveloper, as in image above. For this, click on project properties, and select Libraries from the left hand side of the pane. Click on “Add Library” to add all the available libraries.

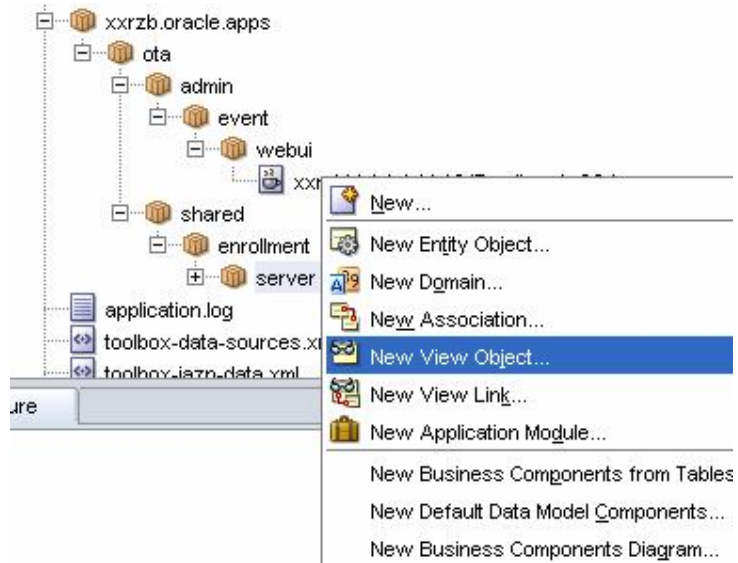
Steps to extend the view Object

1. Create a business component Package named `xxrzb.oracle.apps.ota.shared.enrolment.server`

As per below image, right click on the “server” node to create new view object

<http://oracle.anilpassi.com>

Brief Steps - OA Fwk R12 Extension Example

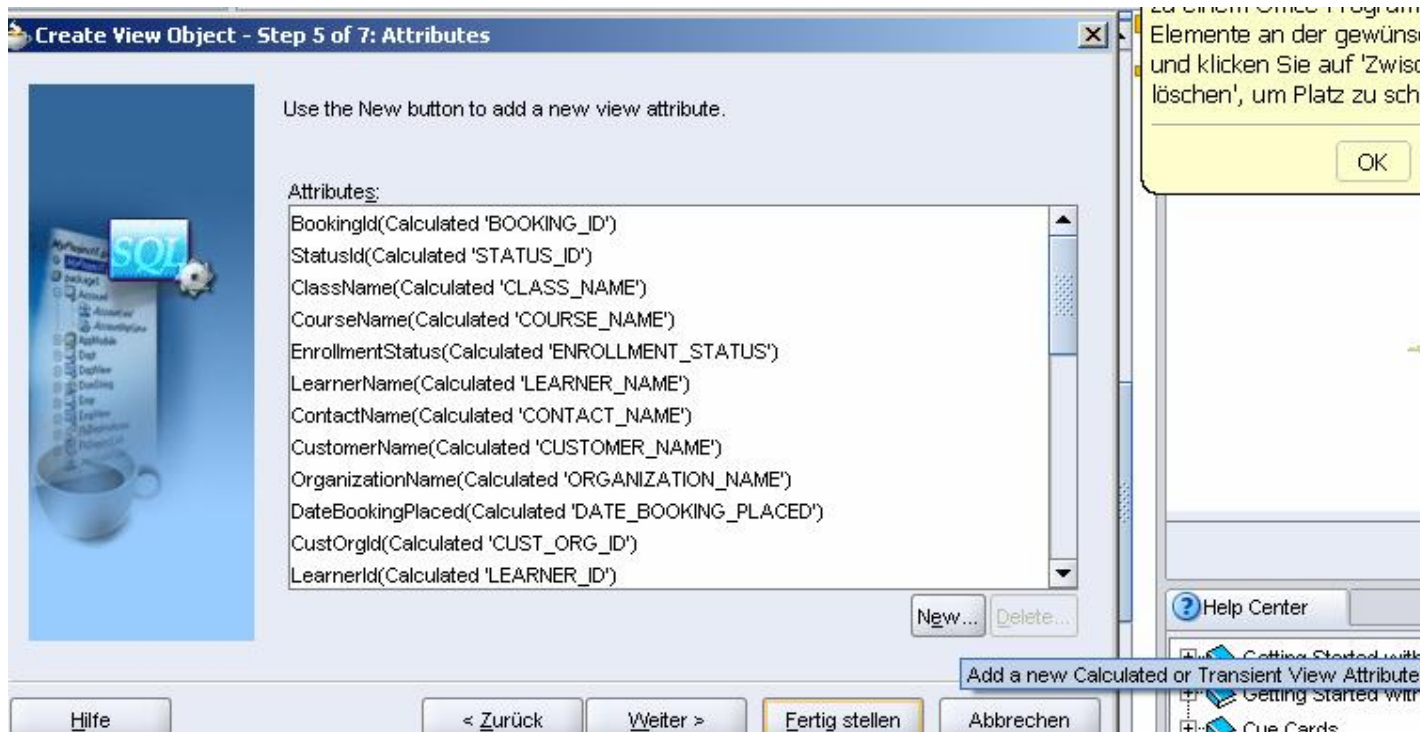


2. This new view object will extend the standard View Object named oracle.apps.ota.shared.enrollment.server.ListOfClsEnrollmentsVO
Name of the new view object will be xxListOfClsEnrollmentsVO

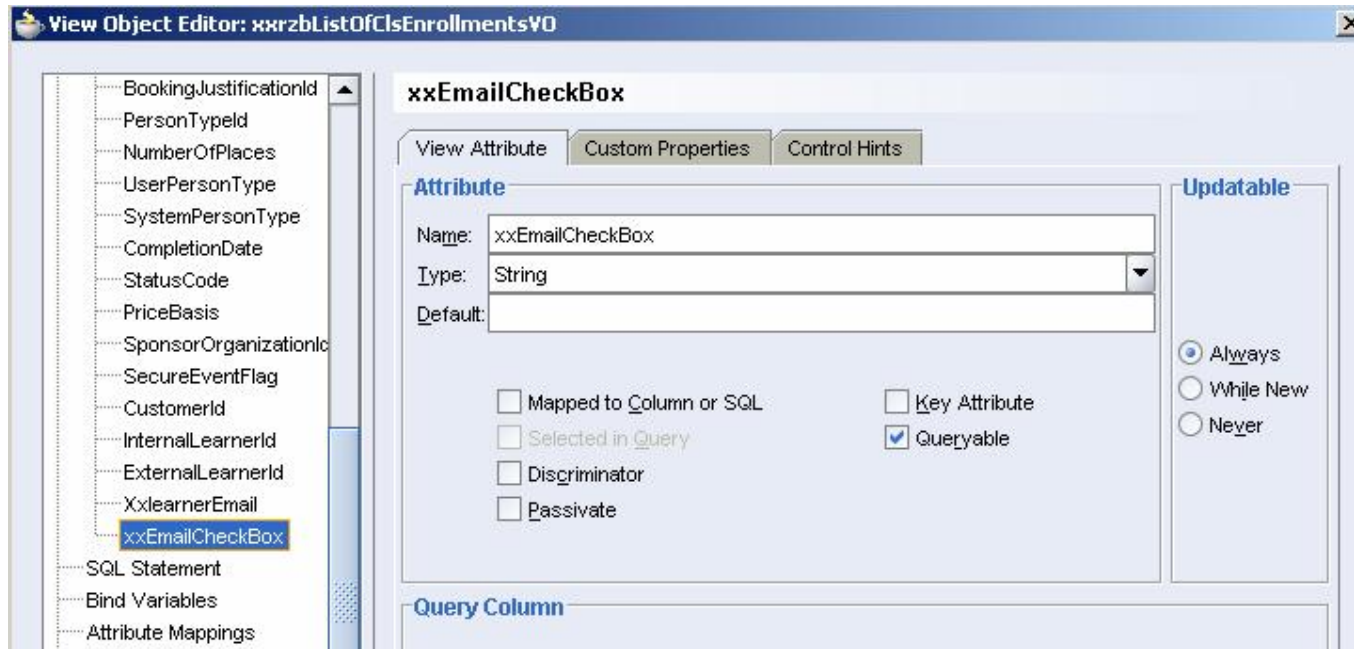


Doing the above will copy the SQL from the parent View Object.
Now we need to add three columns [Two based on PL/SQL Function and one transient]

The pl/sql function can be added directly in SQL.



Click on the “New” button to create a transient attribute required for the checkbox. This is done as below. Ensure that radio button “Updatable” is set to “Always”



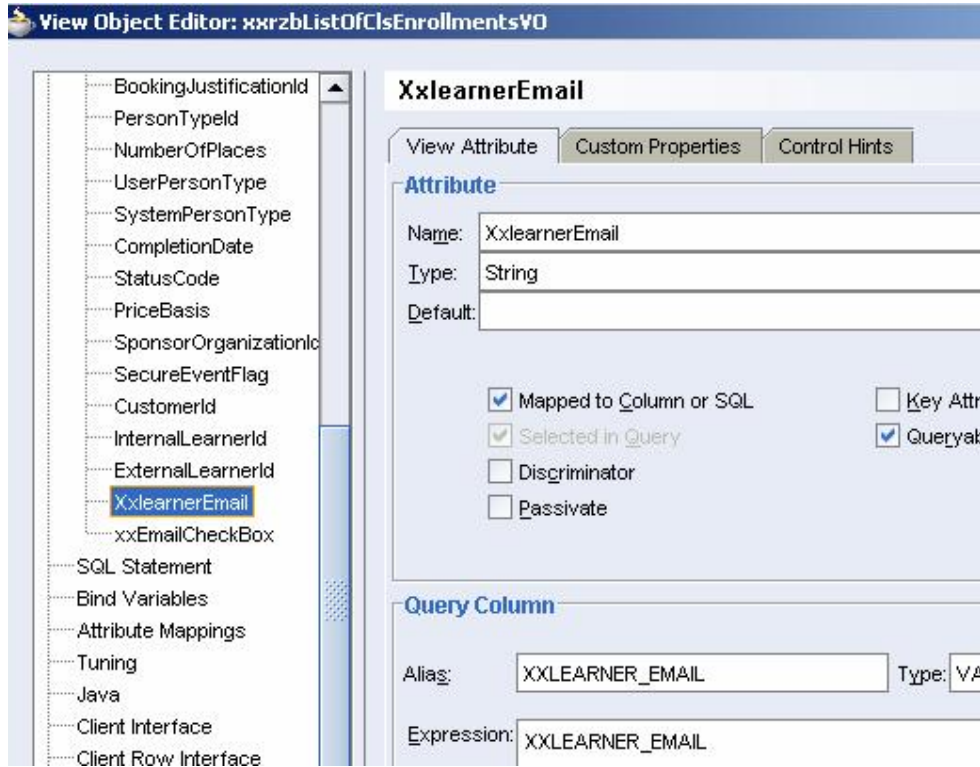
For the view attributes based on pl/sql function, see the below two images

The screenshot shows the Oracle APEX interface. On the left, a vertical list of items is visible, including:

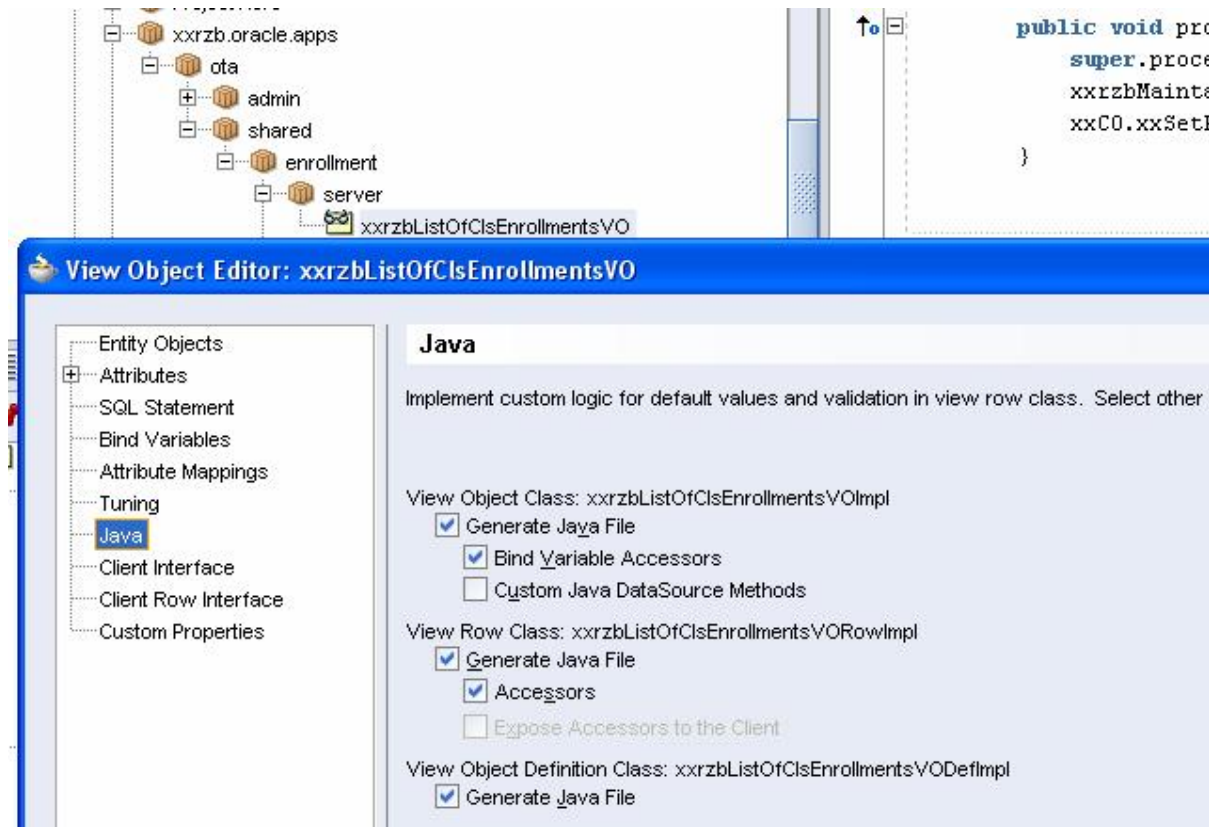
- UserPersonType
- SystemPersonType
- CompletionDate
- StatusCode
- PriceBasis
- SponsorOrganizationId
- SecureEventFlag
- CustomerId
- InternalLearnerId
- ExternalLearnerId
- XlearnerEmail
- xxEmailCheckBox
- SQL Statement** (highlighted)
- Bind Variables

On the right, the 'Query Statement' editor is open, displaying the following SQL code:

```
separately.  
  
Query Statement  
-----  
evt.secure_event_flag,  
tdb.customer_id,  
tdb.delegate_person_id Internal_learner_id,  
tdb.delegate_contact_id external_learner_id,  
xxrzb_get_learner_email(tdb.booking_id) as xxlearner_email  
from  
ota_delegate_bookings tdb,  
ota_events_t1 etl,  
ota_events evt,  
ota_activity_versions_t1 atl,  
ota_booking_status_types_t1 btt,
```

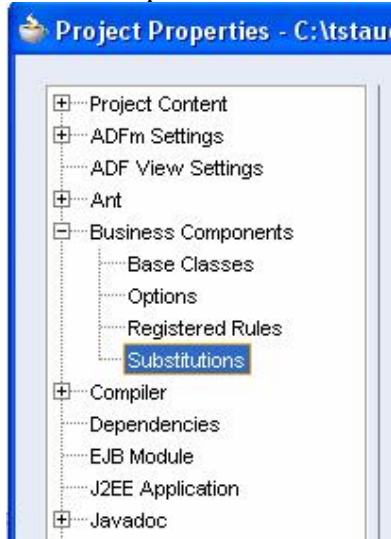


Also, ensure the following checkboxes are checked, so that we can loop through the records on the extended VO easily. This option is available under java in left panel

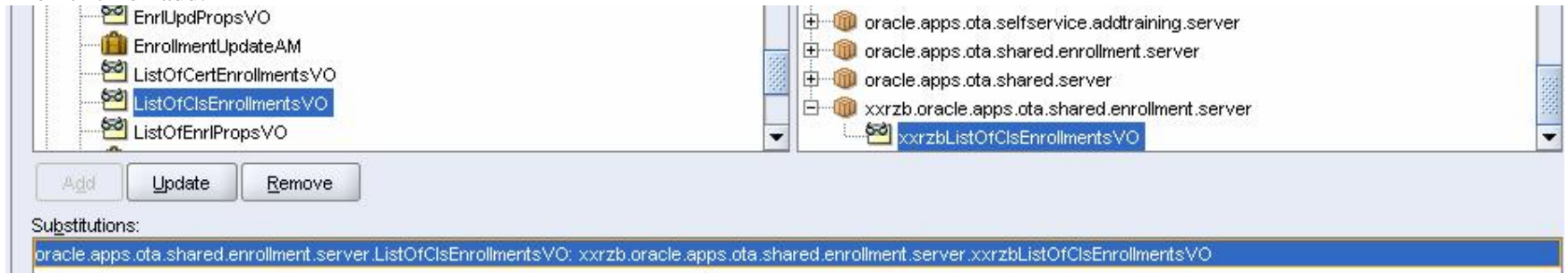


Creation of Substitutions

Double-click on the project, to open the project properties.
Select the option for Substitutions link within the Business Components



Select the original view object on the left hand side, and the new view object on the right hand side.
Then click on add.



<http://oracle.anilpassi.com>

Brief Steps - OA Fwk R12 Extension Example

This will create an entry into the jpx file.

In this case, the jpx file should look like, as below in blue

```
<?xml version='1.0' encoding='windows-1252' ?>
<!DOCTYPE JboProject SYSTEM "jbo_03_01.dtd">

<JboProject
  Name="xxOTA"
  SeparateXMLFiles="true"
  PackageName="" >
  <Substitutes>
    <Substitute      OldName      ="oracle.apps.ota.shared.enrollment.server.ListOfClsEnrollmentsVO"      NewName
="xxrzb.oracle.apps.ota.shared.enrollment.server.xxrzbListOfClsEnrollmentsVO" />
  </Substitutes>
</JboProject>
```

The script for loading the jpx file is

```
MACHINE_NAME=0
PORT_NUMBER=0
```

```
export MACHINE_NAME
export PORT_NUMBER
```

```
MACHINE_NAME=oracle01.focusthread.com
PORT_NUMBER=1571
```

"Echo load substitution"

```
adjava \
oracle.jrad.tools.xml.importer.JPXImporter $c_rzb_top/oaf/xxRzbSubstituteListOfClsEnrollmentsVO.jpx \
-username apps \
-password apps \
```

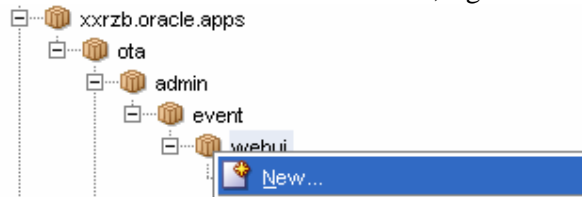
-dbconnection

```
"(description=(address_list=(address=(community=tcp.world)(protocol=tcp)(host=$MACHINE_NAME)(port=$PORT_NUMBER)))(connect_data=(sid=$TWO_TASK)))"
```

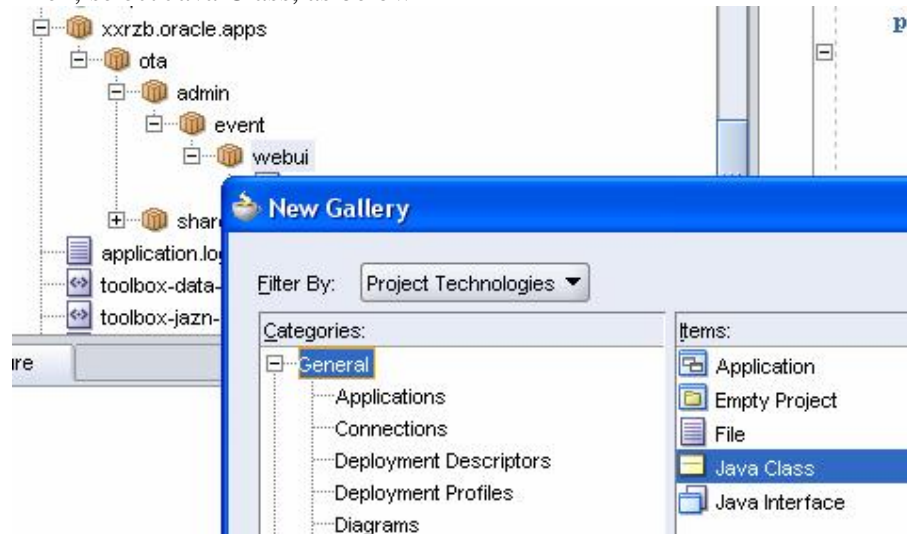
Steps to extend Controller

The extended controller was created in package xxrzb.oracle.apps.ota.admin.event.webui

To create the extension of controller, right click on webui in the tree, underneath xxrzb, and select "New"



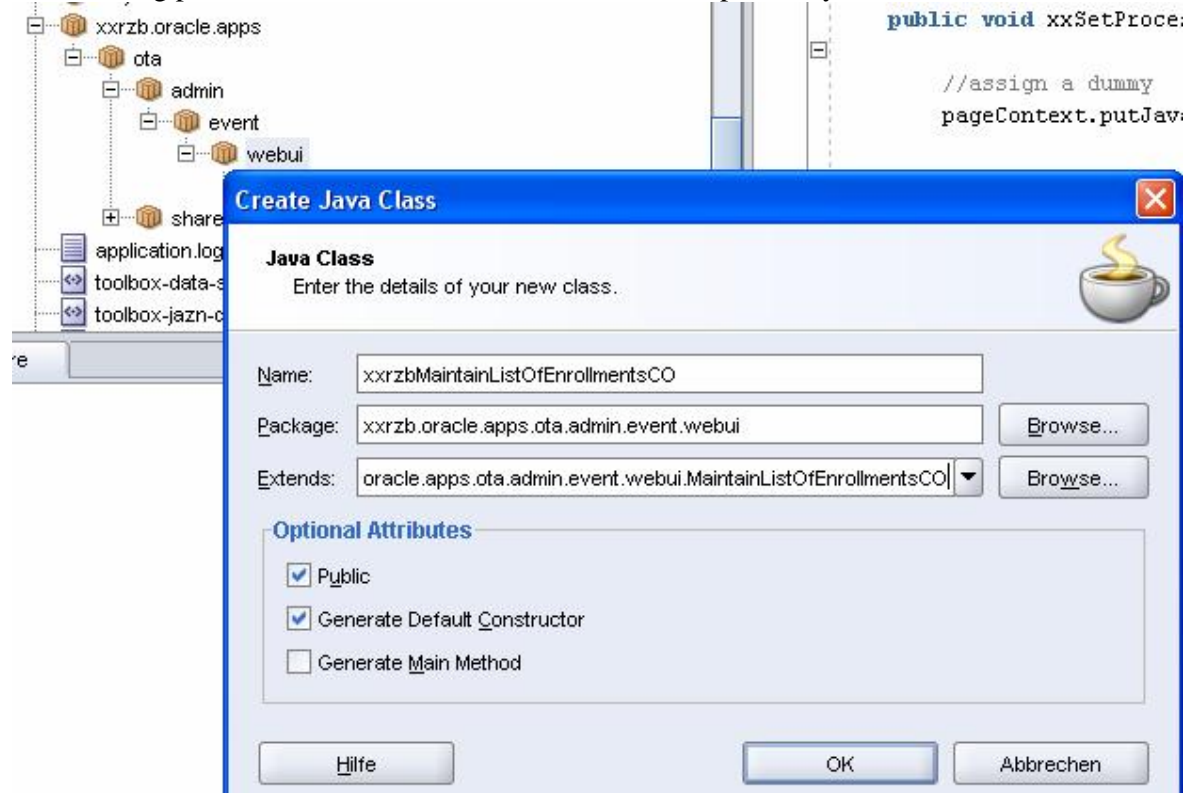
Then, select Java Class, as below



And then enter the new Class Name, the PackageName[starting with xx], and the standard class name in field “Extends”

You may also click on browse button to select the Standard Controller Class

Note:- Using personalization, the standard class must be replaced by custom class, at Site level.



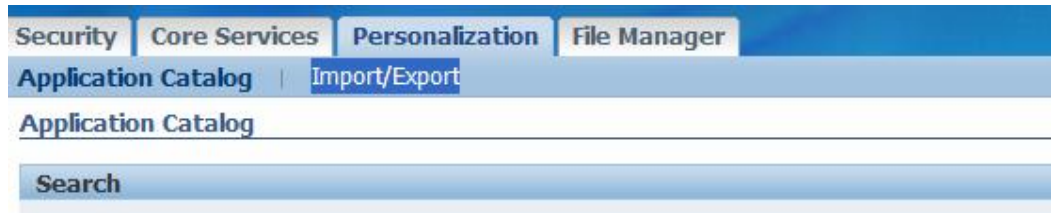
Steps to export personalizations

Navigate to responsibility “Functional Administrator”

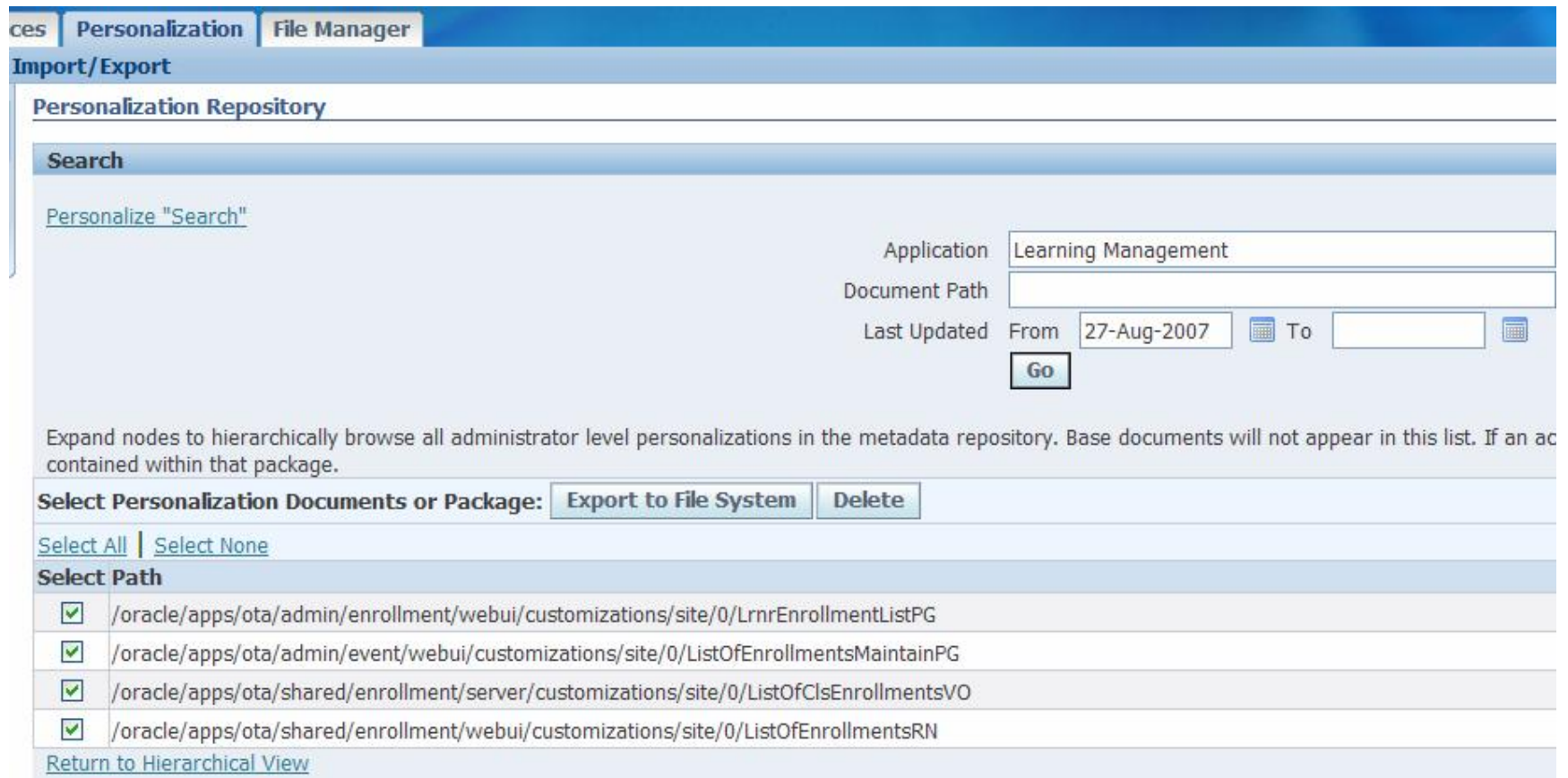
And within that, click Personalization and then click Import/Export

<http://oracle.anilpassi.com>

Brief Steps - OA Fwk R12 Extension Example



Search for personalizations based on Application name and the Last Updated field.
Click on Go, and select the personalizations that you wish to Export to file system
The file tree will be created in a directory on server. Name of the directory is mentioned in profile FND%Root%



Click on "Export to File System"

In this case, as per the profile option, the directory is "/tmp/custdocs"

Create the tar file using command below

```
tar -cvf xxEmailPersonalize.tar ./oracle/*
```

This file xxEmailPersonalize.tar can be FTPed to the new environment, from where you can unzip this by command

<http://oracle.anilpassi.com>

Brief Steps - OA Fwk R12 Extension Example

```
tar -xvf xxEmailPersonalize.tar ./
```

IMPORTANT:- Ensure that all the files from myclasses/xxrzb directory get copied to \$JAVA_TOP

The tar file with personalizations is also FTPed to C:\anilpassi\JDev_for_R12\jpxScripts

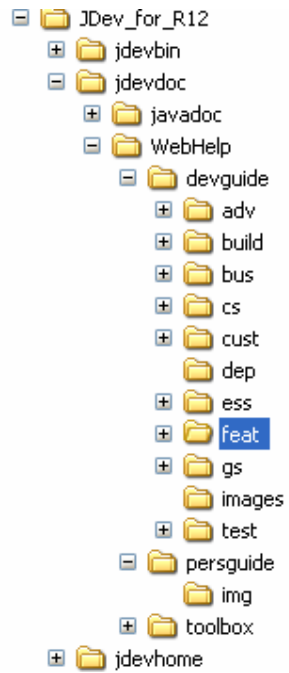
List of actions before user sign-off

1. Create proper pl/sql functions that will return the email addresses, for both the Learner and the Supervisor
2. Edit the extended view object to make calls to these pl/sql functions
3. The controller code needs to be enhanced to calculate email string for Supervisor as well
4. An additional Button must be added, for sending emails to Supervisors as well
5. If required, extend the controller on other pages.
6. Deploy the changes to extended VO and extended CO and Personalization

<http://oracle.anilpassi.com>

Brief Steps - OA Fwk R12 Extension Example

Useful documentation:



Do you want OA Framework training in London?
Express your interest by writing to anilpassi@gmail.com