Oracle Apps

Oracle Application
Framework (OAF)

Objectives

- At the end of this session, you will be able to:
  - Understand the concept of Oracle Application Framework (OAF) and its Architecture.
  - Understand the concept of Model View Controller (MVC).
  - Learn about Jdeveloper.
  - Understand the concept of Model.
  - Understand the concept of BCFJ Objects.
  - Understand the concept of View.
  - Understand the concept of Controller.
  - Learn about Validations.
  - Learn about Extensions and its types.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2

Oracle Application Framework

- Oracle Application Framework (OA Framework or OAF) is a framework used for application development within the Oracle E-Business Suite (EBS).
- The framework is also available to customers for personalization, customizations and custom-application development.
- It is the Oracle Applications development and deployment platform for HTML-based business applications.
- The OA Framework is a Model-view-controller (MVC) framework built using Java EE technologies.
- These pages are designed to be familiar to web-based users, and easy to deploy within a web browser (requiring no plugin or download, unlike Oracle Forms).

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3

Oracle Application Framework (Contd.)

- The OA Framework helps to create self Service pages in Oracle EBS.
- It helps in controlling the flow of the application.
- The OA Framework is J2EE based but it also supports various standards like HTML, XML, SQL, JSP.

- OAF Uses: The necessary to use OAF in Oracle applications is Integration, Security, and Customization.
  - Integration: Registration of Forms is easy.
  - Security: Flexibility reasons if we are using third party we don't know when it is going to fail.
  - Customization: If we are integrating with third party customization is tough where as using OAF customization is little bit easier when comparing to Other Third party customization.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

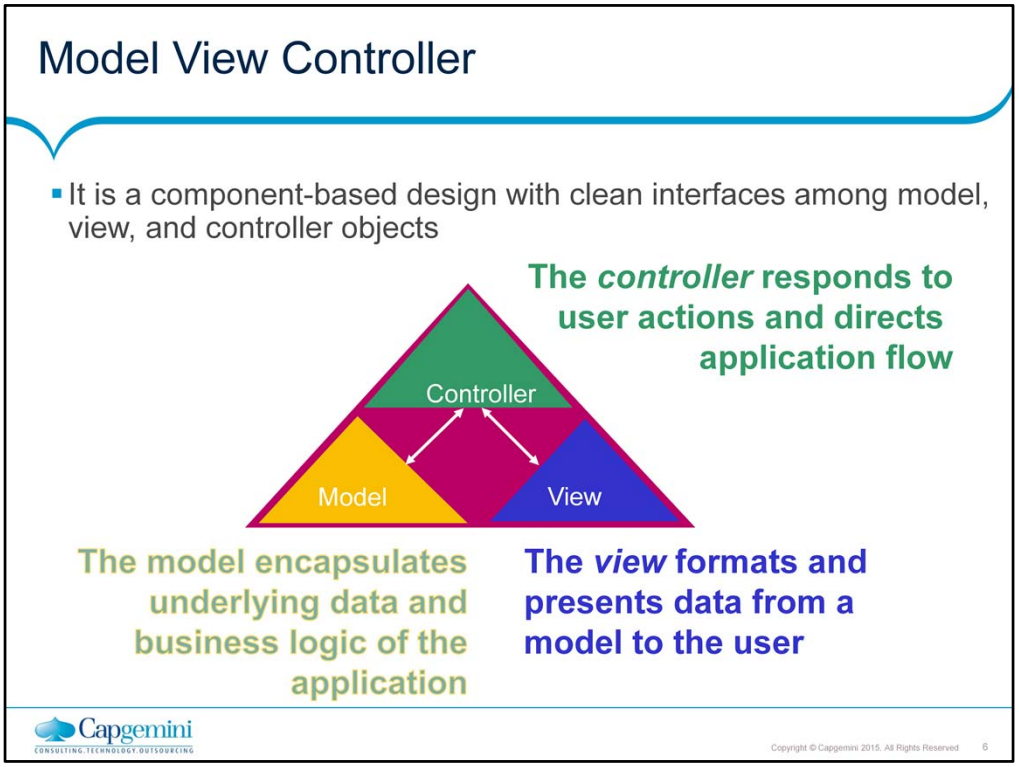
 4

Oracle Application Framework (Contd.)

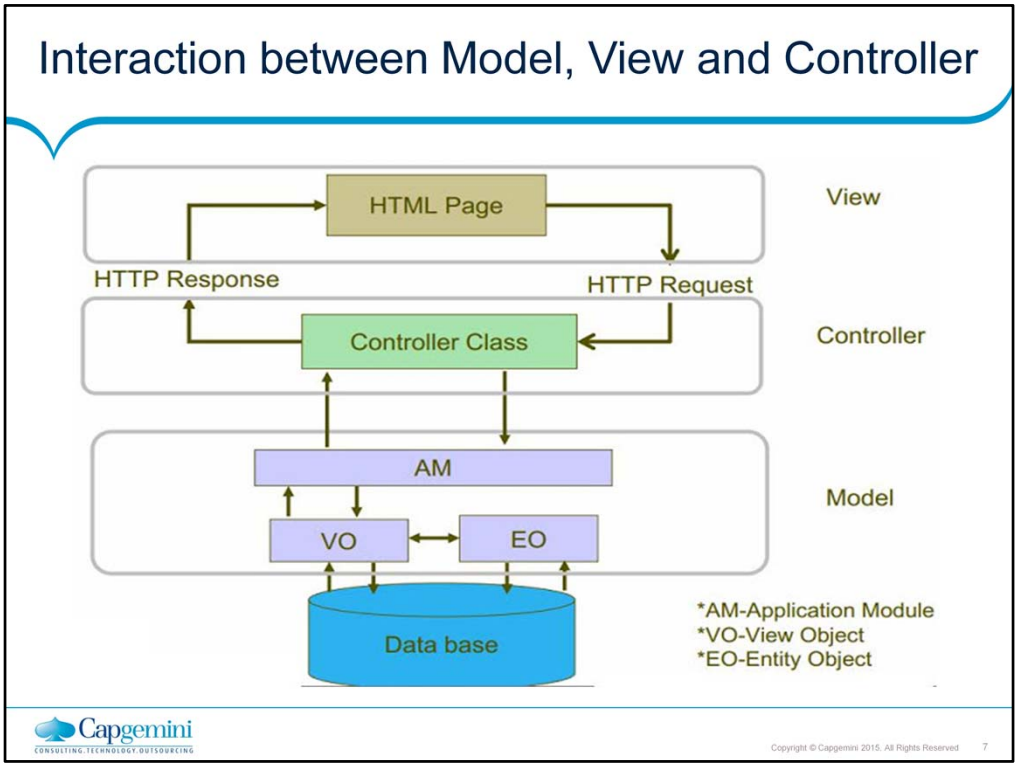
- OAF Advantages & Disadvantage: The advantages of OAF are as follows:
  - End user Productivity
  - Enterprise Grade Performance and Scalability
  - Highly extensible Architecture
  - Browser Look and Feel of all the pages in application.
  - Open Standards such as XML, HTML, Java, JSP, SQL, and Web Services.
  - Application Customizability.
  - Developer Productivity.
- The disadvantages of OAF are:
  - Cannot see the layout at design time.
  - OAF Pages are integrated / compatible only with the Oracle Apps.
  - Drag and Drop options are not available.
  - More R&D is required to design a form layout which is time consuming.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 5



Add the notes here.


Interaction between Model, View and Controller
HTML Page
View
HTTP Response
HTTP Request
Controller Class
Controller
AM
Model
VO
EO
Data base
*AM-Application Module
*VO-View Object
*EO-Entity Object
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
7

OAF Architecture

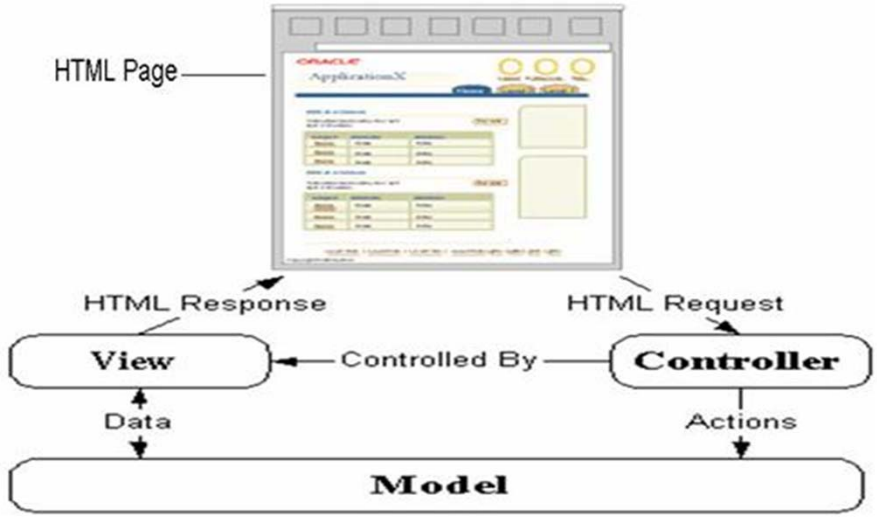
- Oracle Application Framework (OAF) is an architecture for creating web based front end pages and J2EE type of applications within the Oracle EBS ERP platform.
- In order to develop and maintain OAF functionality, Oracle's JDeveloper tool is used.
- OAF is based on J2EE technology called BC4J (Business Components for Java).
- As per the MVC architecture, in OAF, the XML Page forms the View, the JAVA based controller class forms the controller and the Application Module along with View Objects (VO) and Schema Objects (EO) forms the Model

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8

Add the notes here.

OAF Architecture (Contd.)



 9

JDeveloper

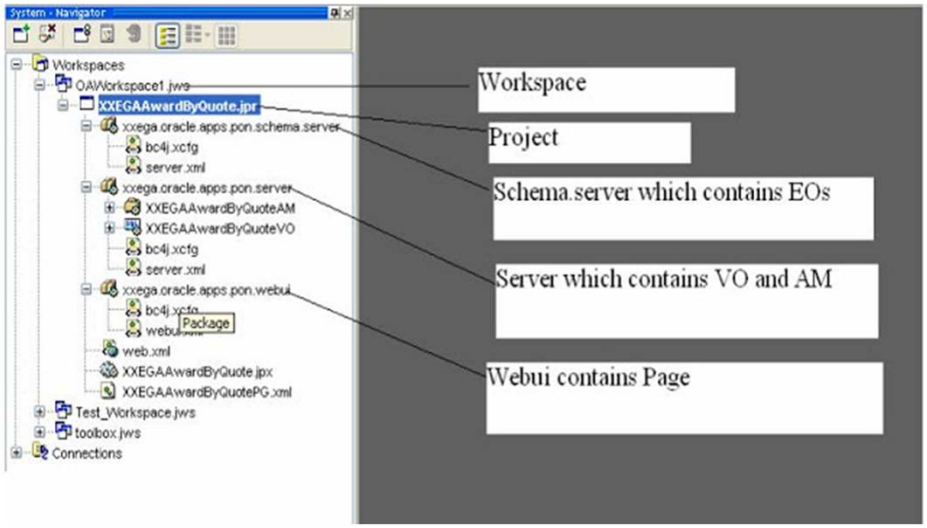
- JDeveloper is a freeware IDE supplied by Oracle Corporation. It offers features for development in Java, XML, SQL and PL/SQL, HTML, JavaScript, BPEL and PHP.
- JDeveloper covers the full development lifecycle from design through coding, debugging, optimization and profiling to deploying.
- With JDeveloper, Oracle has aimed to simplify application development by focusing on providing a visual and declarative approach to application development in addition to building an advanced coding-environment.
- Oracle JDeveloper integrates with the Oracle Application Development Framework(Oracle ADF) - an end-to-end Java EE-based framework that further simplifies application development.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

Add the notes here.

Jdeveloper Structure

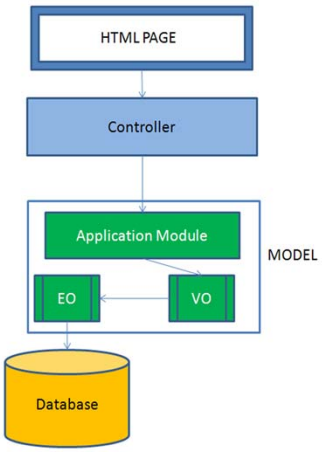


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11

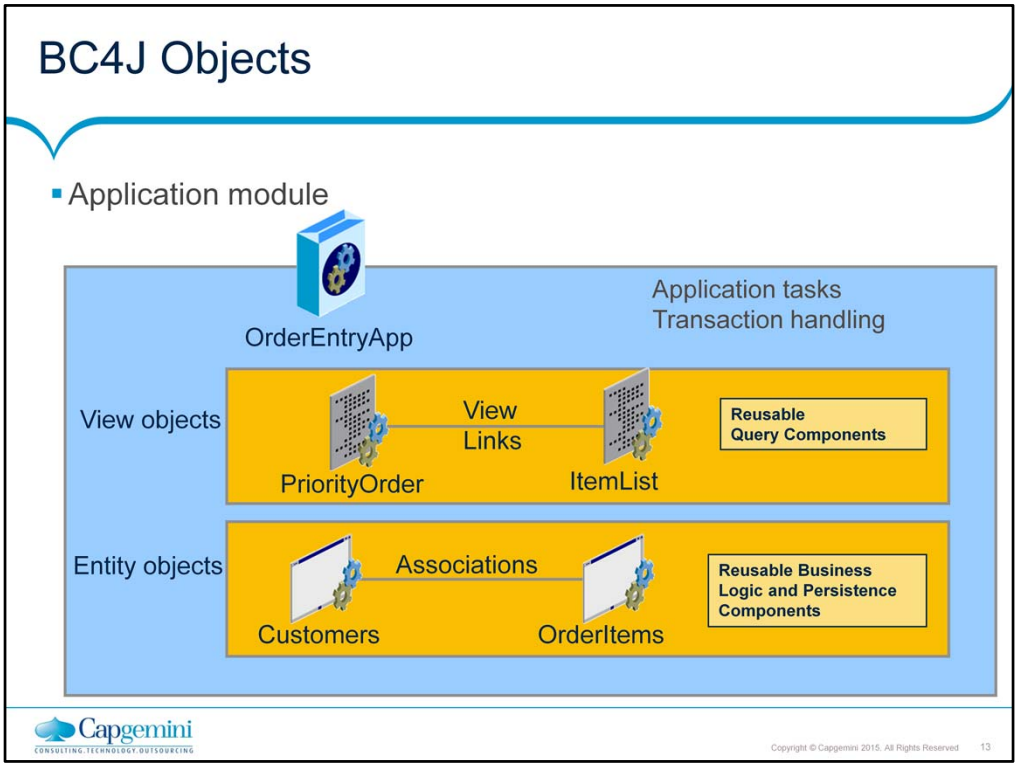
MODEL: Business Components for Java (BC4J)

- Model contains the components which handles data directly from database.
- There are three basic component classes:
  - Application Module – a container for related BC4J objects
  - Entity Objects (EO) – encapsulate business rules
  - View Objects (VO) – present data to the Framework page



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

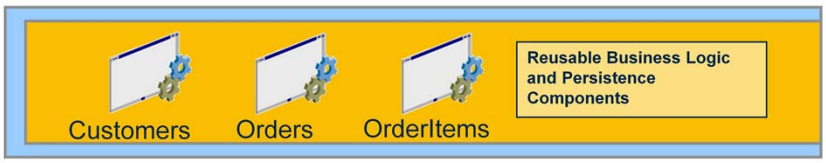
 12


BC4J Objects
Application module
OrderEntryApp
Application tasks
Transaction handling
View objects
View
Links
PriorityOrder
ItemList
Reusable
Query Components
Entity objects
Associations
Customers
OrderItems
Reusable Business
Logic and Persistence
Components
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
13

Entity Objects

- The Entity Objects are used if one wishes to do some insert/update operations.
- It maps to a database table or other data source
- Each entity object instance represents a single row
- It contains attributes representing database columns
- It supports Fundamental BC4J object through which all inserts/updates/deletes interact with the database
- It supports Business logic and validation related to a table
- It can contain Custom Business Methods

Entity objects



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 14

View Objects

- In cases where one just want some data just for view purpose and want to show it on the page or use the values for some other purpose then one uses View objects.
- It is used for joining, filtering, projecting, and sorting your business data.
- It can be based on any number of entity objects.
- It can also be constructed from a SQL statement (Expert Mode).
- It contains only the attributes required for a specific purpose i.e, specific to a particular UI.

View objects

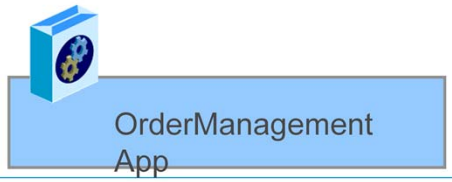
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 15

Application Module

- It is a very important component for the package, it is the one which governs the entire session pool, access to database and the business logic components, every page should be attached to some Application Module.
- It defines the logical data model and writes business methods needed to perform application tasks.
- It handles transactions objects.
- It is a Container for view objects and view links.
- It Initializes and performs view object query.

Application Module

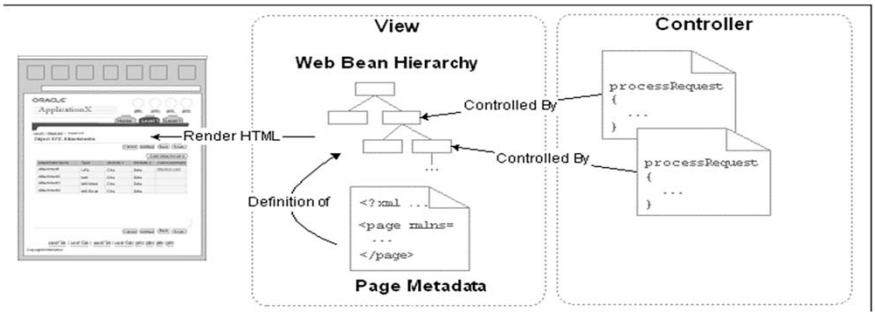


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16

Page and Region

- It renders as normal HTML:



- In the middle tier page is implemented in memory as a hierarchy of Java beans.
- Each UI element that renders in the page (buttons, a table, tabs, etc) corresponds to one or more web beans in the hierarchy

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

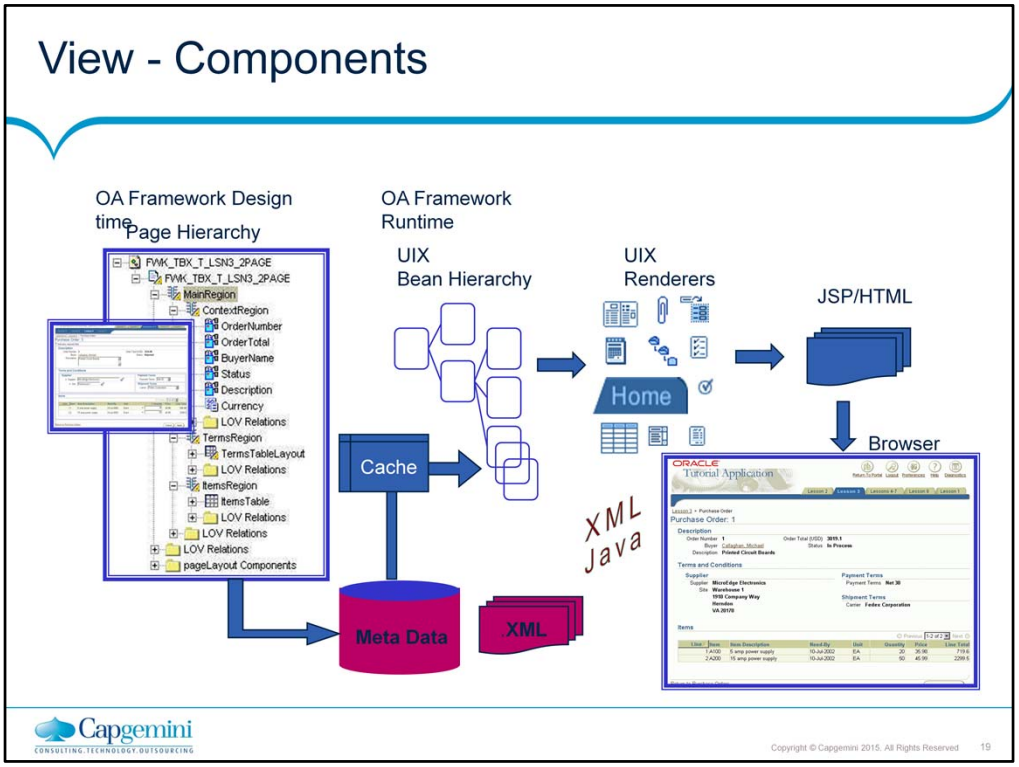
 17

VIEW

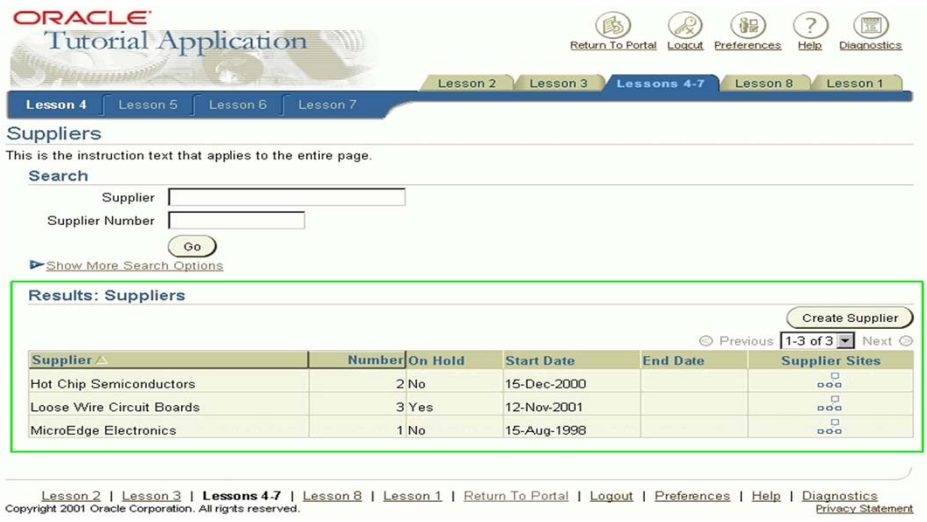
- View is the actual output of OAF pages.
- The View formats the data and presents the data to the user.
- In OAF, View is implemented using the UIX.
- UIX uses XML to describe the components and hierarchy that make up an application page.
- UIX also provides runtime capabilities to translate that metadata into HTML output so that it can be shown on a Browser or a mobile device.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 18


View - Components
OA Framework Design time
Page Hierarchy
FWK_TBX_T_LSN3_2PAGE
FWK_TBX_T_LSN3_2PAGE
MainRegion
ContextRegion
OrderNumber
OrderTotal
BuyerName
Status
Description
Currency
LOV Relations
TermsRegion
TermsTableLayout
LOV Relations
ItemsRegion
ItemsTable
LOV Relations
LOV Relations
LOV Relations
pageLayout Components
OA Framework Runtime
UIX Bean Hierarchy
UIX Renderers
JSP/HTML
Home
Browser
Cache
XML
Java
Meta Data
XML
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
19

View: A Basic Framework Example

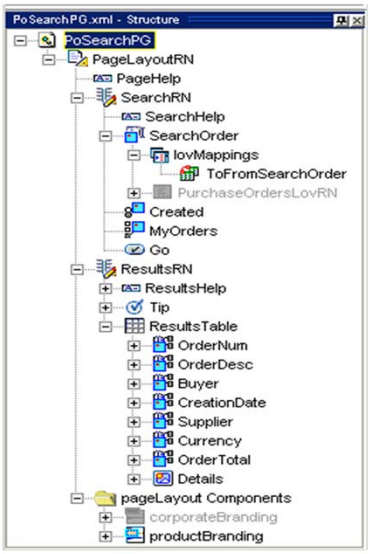


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20

View: Page Hierarchy

- This is the page structure as seen in OA Extension at design time.
- The Framework uses the order of the items to determine their position within a page or region UI at runtime.



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 21

CONTROLLER

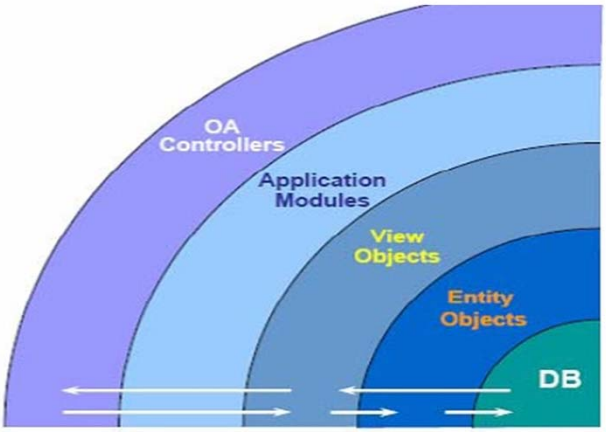
- When user clicks a button, or performs certain action what responses should be triggered is coded in the Controller.
- Controller handles all the user actions done on the page.
- It responds to user action and direct application flow.
- Controller will take care of the web browser activities.
- Controller responds to user actions and directs the application flow.
- It provides the wiring between the UIX web bean and the middle-tier.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22

Controller (Contd.)

- Model objects like EO and VO can't be accessed directly from the Controller Class, except AM



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 23

Validations

- Client Side Validations.
  - Required Property.
  - Data Types.
  - Formats.
- Server Side Validations.
  - Attribute level.
  - Entity Objects (EO) level
  - View Objects (VO) level.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 24

Extensions and its Types

- We use extensions if we want to do any changes in the already shipped oracle E-Business suite.
- The main purpose we use extensions is to make changes in the already well designed Oracle E-Business suite pages.
- The different types of extensions in OAF (Oracle Apps Framework) are:
  - BC4J Extensions
    - View Object (VO) Extension
    - Application Module (AM) Extension
    - Entity Object (EO) Extension
  - Controller (CO) Extension

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

Summary

- In this session, we have covered:
  - Oracle Application Framework and its Architecture
  - Model View Controller
  - Jdeveloper
  - Model
  - BC4J Objects
  - View
  - Controller
  - Validations
  - Extensions and its Types

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 26

Add the notes here.

```
declare
        a emp%rowtype;
begin
        select * into a from emp where empno=2100;
exception
        when no_data_found then
        if sql%notfound then
                dbms_output.put_line('true');
        elsif sql%notfound=false then
                dbms_output.put_line('false');
        elsif sql%notfound is null then
                dbms_output.put_line('null');
        end if;
if sql%found then
                dbms_output.put_line('true');
        elsif sql%found=false then
                dbms_output.put_line('false');
        elsif sql%found is null then
                dbms_output.put_line('null');
        end if;
 end;
 /
```

Considering that 2100 does not exist

Output :

true

False

```
declare
        a number;
begin
        select sum(sal) into a from emp where deptno=1000;
        if sql%notfound then
                dbms_output.put_line('true');
        elsif sql%notfound=false then
                dbms_output.put_line('false');
        elsif sql%notfound is null then
                dbms_output.put_line('null');
        end if;
        if sql%found then
                dbms_output.put_line('true');
        elsif sql%found=false then
                dbms_output.put_line('false');
        elsif sql%found is null then
                dbms_output.put_line('null');
        end if;
 exception
        when no_data_found then
                if sql%notfound then
                        dbms_output.put_line('etrue');
                elsif sql%notfound=false then

        dbms_output.put_line('efalse');
                elsif sql%notfound is null then
                        dbms_output.put_line('enull');
                end if;
if sql%found then
                        dbms_output.put_line('etrue');
                elsif sql%found=false then

        dbms_output.put_line('efalse');
                elsif sql%found is null then
                        dbms_output.put_line('enull');
                end if;
 end;
/
```

(regardless of whether deptno=1000 exists or not)

Output :

```
false
true
```

```
declare
        a number;
begin
        select sal into a from emp where deptno=10;
exception
        when too_many_rows then
                if sql%notfound then
                        dbms_output.put_line('etrue');
                elsif sql%notfound=false then
                        dbms_output.put_line('efalse');
                elsif sql%notfound is null then
                        dbms_output.put_line('enull');
                end if;
if sql%found then
                        dbms_output.put_line('etrue');
                elsif sql%found=false then
                        dbms_output.put_line('efalse');
                elsif sql%found is null then
                        dbms_output.put_line('enull');
                end if;
dbms_output.put_line('number of rows is :'||sql%rowcount);
 end;
/
```

Output :

efalse

etrue

number of rows is :1

Use of collection as a target :

```
declare
        type x is table of emp.sal%type;
        a x;
begin
        select sal bulk collect into a from emp where deptno=18;
        if sql%notfound then
                dbms_output.put_line('true');
        elsif sql%notfound=false then
                dbms_output.put_line('false');
        elsif sql%notfound is null then
                dbms_output.put_line('null');
        end if;
if sql%found then
                dbms_output.put_line('true');
        elsif sql%found=false then
                dbms_output.put_line('false');
        elsif sql%found is null then
                dbms_output.put_line('null');
        end if;
dbms_output.put_line('number of rows is :'||sql%rowcount);
 exception
        when too_many_rows then
                if sql%notfound then
                        dbms_output.put_line('etrue');
                elsif sql%notfound=false then
                        dbms_output.put_line('efalse');
                elsif sql%notfound is null then
                        dbms_output.put_line('enull');
                end if;
if sql%found then
                        dbms_output.put_line('etrue');
                elsif sql%found=false then
                        dbms_output.put_line('efalse');
                elsif sql%found is null then
                        dbms_output.put_line('enull');
                end if;
dbms_output.put_line('number of rows is :'||sql%rowcount);
 end;
/
```

because of use of collection as a target.

Output :

```
false
true
number of rows is :3
```

For imlicit cursors :

```
declare
begin
        dbms_output.put_line('before DML');
        if sql%isopen=true then
                dbms_output.put_line('cursosr is open');
        elsif sql%isopen=false then
                dbms_output.put_line('cursosr is closed');
        else
                dbms_output.put_line('%isopen is null');
        end if;

if sql%found=true then
                dbms_output.put_line('%found is true');
        elsif sql%found=false then
                dbms_output.put_line('%found is false');
        else
                dbms_output.put_line('%found is null');
        end if;

if sql%notfound=true then
                dbms_output.put_line('%notfound is true');
        elsif sql%notfound=false then
                dbms_output.put_line('%notfound is
false');
        else
                dbms_output.put_line('%notfound is null');
        end if;

if sql%rowcount>0 then
                dbms_output.put_line('Found
'||sql%rowcount || 'rows');
        elsif sql%rowcount=0 then
                dbms_output.put_line('cursosr found 0
rows');
        else
                dbms_output.put_line('%rowcount is null');
        end if;
```

```
delete from emp where deptno=10;
                dbms_output.put_line('after DML');
                if sql%isopen=true then
                                dbms_output.put_line('cursosr is open');
                elsif sql%isopen=false then
                                dbms_output.put_line('cursosr is closed');
                else
                                dbms_output.put_line('%isopen is null');
                end if;

if sql%found=true then
                                dbms_output.put_line('%found is true');
                elsif sql%found=false then
                                dbms_output.put_line('%found is false');
                else
                                dbms_output.put_line('%found is null');
                end if;

if sql%notfound=true then
                                dbms_output.put_line('%notfound is true');
                elsif sql%notfound=false then
                                dbms_output.put_line('%notfound is
false');
                else
                                dbms_output.put_line('%notfound is null');
                end if;

if sql%rowcount>0 then
                                dbms_output.put_line('Found
'||sql%rowcount || 'rows');
                elsif sql%rowcount=0 then
                                dbms_output.put_line('cursosr found 0
rows');
                else
                                dbms_output.put_line('%rowcount is null');
                end if;
end;
/
```

Output :

before DML

cursosr is closed

%found is null

%notfound is null

%rowcount is null

after DML

cursosr is closed

%found is true

%notfound is false

Found 3rows

For explicit cursors :

```
declare
        empdata emp%rowtype;
        cursor c1 is select * from emp where deptno=10;
begin
dbms_output.put_line('before opening cursor');
        if c1%isopen=true then
                dbms_output.put_line('cursosr is
open');
        elsif c1%isopen=false then
                dbms_output.put_line('cursosr is
closed');
        else
                dbms_output.put_line('%isopen is
null');
        end if;
--if c1%found=true then
                --  dbms_output.put_line('%found is
true');
-- elsif c1%found=false then
                --  dbms_output.put_line('%found is
false');
-- else
                --  dbms_output.put_line('%found is
null');
        -- end if;
```

```
-- if c1%notfound=true then
                                --  dbms_output.put_line('%notfound is
true');
                --elsif c1%notfound=false then
                                -- dbms_output.put_line('%notfound is
false');
                --else
                                -- dbms_output.put_line('%notfound is
null');
-- end if;
-- if c1%rowcount>0 then
                                --   dbms_output.put_line('Found
'||c1%rowcount || 'rows');
                -- elsif c1%rowcount=0 then
                                --   dbms_output.put_line('cursosr
found 0 rows');
-- else
                                --  dbms_output.put_line('%rowcount is
null');
                --end if;
open c1;
dbms_output.put_line('after opening cursor');
if c1%isopen=true then
                                dbms_output.put_line('cursosr is
open');
                elsif c1%isopen=false then
                                dbms_output.put_line('cursosr is
closed');
                else
                                dbms_output.put_line('%isopen is
null');
                end if;
```

```
if c1%found=true then
                dbms_output.put_line('%found is true');
        elsif c1%found=false then
                dbms_output.put_line('%found is
false');
        else
                dbms_output.put_line('%found is null');
        end if;

if c1%notfound=true then
                dbms_output.put_line('%notfound is
true');
        elsif c1%notfound=false then
                dbms_output.put_line('%notfound is
false');
        else
                dbms_output.put_line('%notfound is
null');
        end if;

if c1%rowcount>0 then
                dbms_output.put_line('Found
'||c1%rowcount || 'rows');
        elsif c1%rowcount=0 then
                dbms_output.put_line('cursosr found 0
rows');
        else
                dbms_output.put_line('%rowcount is
null');
        end if;
```

```
fetch c1 into empdata;
dbms_output.put_line('after fetching from cursor');
if c1%isopen=true then
                        dbms_output.put_line('cursosr is
open');
            elsif c1%isopen=false then
                        dbms_output.put_line('cursosr is
closed');
            else
                        dbms_output.put_line('%isopen is
null');
            end if;
if c1%found=true then
                        dbms_output.put_line('%found is true');
            elsif c1%found=false then
                        dbms_output.put_line('%found is
false');
            else
                        dbms_output.put_line('%found is null');
            end if;

if c1%notfound=true then
                        dbms_output.put_line('%notfound is
true');
            elsif c1%notfound=false then
                        dbms_output.put_line('%notfound is
false');
            else
                        dbms_output.put_line('%notfound is
null');
            end if;
```

```
if c1%rowcount>0 then
                    dbms_output.put_line('Found
'||c1%rowcount || 'rows');
          elsif c1%rowcount=0 then
                    dbms_output.put_line('cursosr found
0 rows');
          else
                    dbms_output.put_line('%rowcount is
null');
          end if;
 end;
/
```

OUTPUT :

```
before opening cursor
cursosr is closed
after opening cursor
cursosr is open
%found is null
%notfound is null
cursosr found 0 rows
after fetching from cursor
cursosr is open
%found is true
%notfound is false
Found 1rows
```

Except isopen all other 3 attributes available only after the cursor is opened.