

# Oracle JHeadstart

V 2.1 Julio 2013

1. Qué es JHeadstart
2. Service Definition
3. JHeadstart Generator
4. Ejecución de aplicaciones generadas
5. Seguridad
6. Menús
7. Manejo de errores

# 1 Qué es JHeadstart



## Introducción

---

- Es una desarrollado y mantenido por Oracle Consulting.
- Es un development toolkit construido sobre ADF.
- Genera aplicaciones web ADF, con funcionalidad transaccional.
- Genera GUI tales como wizards, shuttles, LOV, búsqueda avanzada y seguridad basada en roles.
- Está totalmente integrado a JDeveloper.
- La generación se hace a partir de metadata (XML) sin requerir código java.

# 1 Qué es JHeadstart

## Beneficios

---

- Alta productividad.
- Facilita la transición JEE.
- Look and feel consistente.
- Integración con Oracle Designer.
- Metadata Technology-agnostic.
- Preservación de cambios post-generación.

# 1 Qué es JHeadstart

Como usar JHeadstart

- Arquitectura soportada

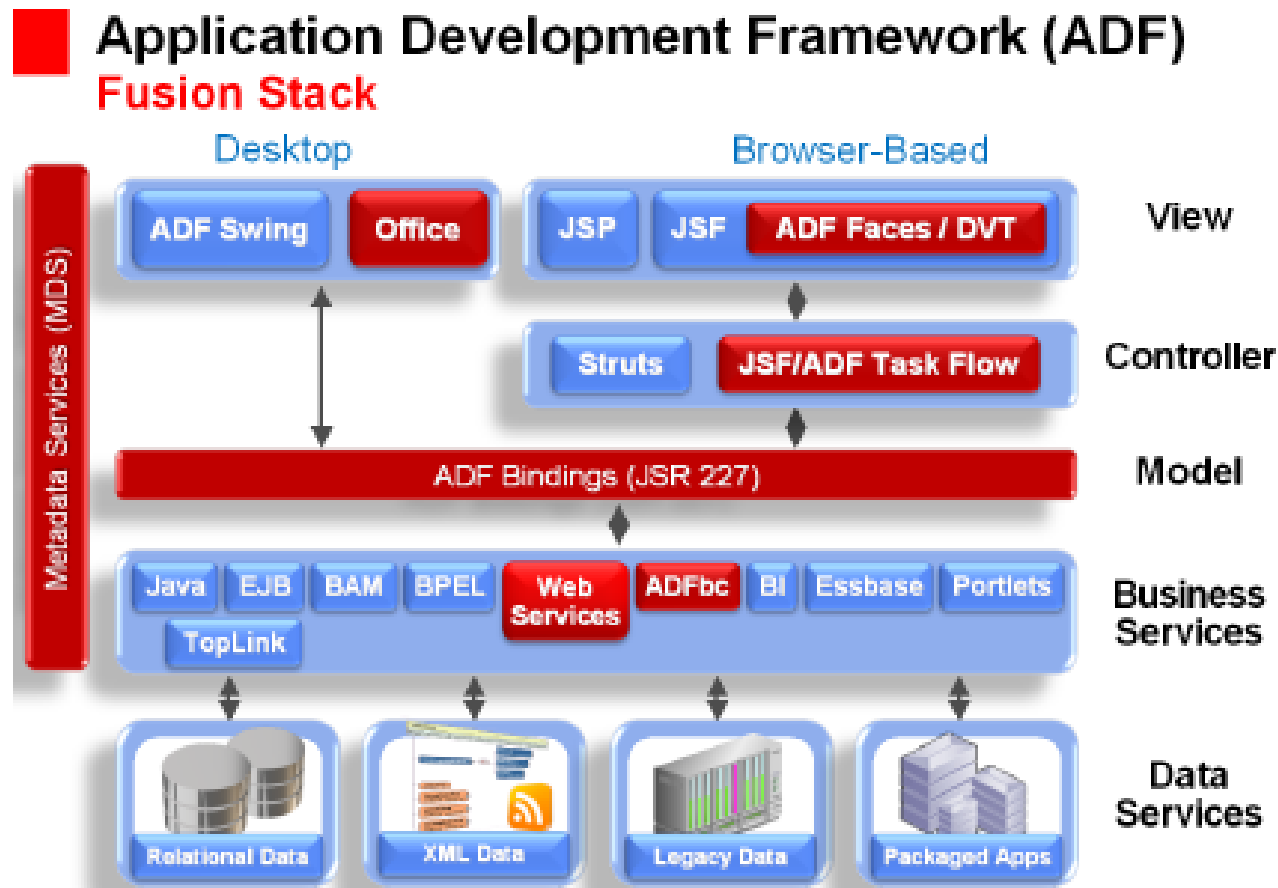
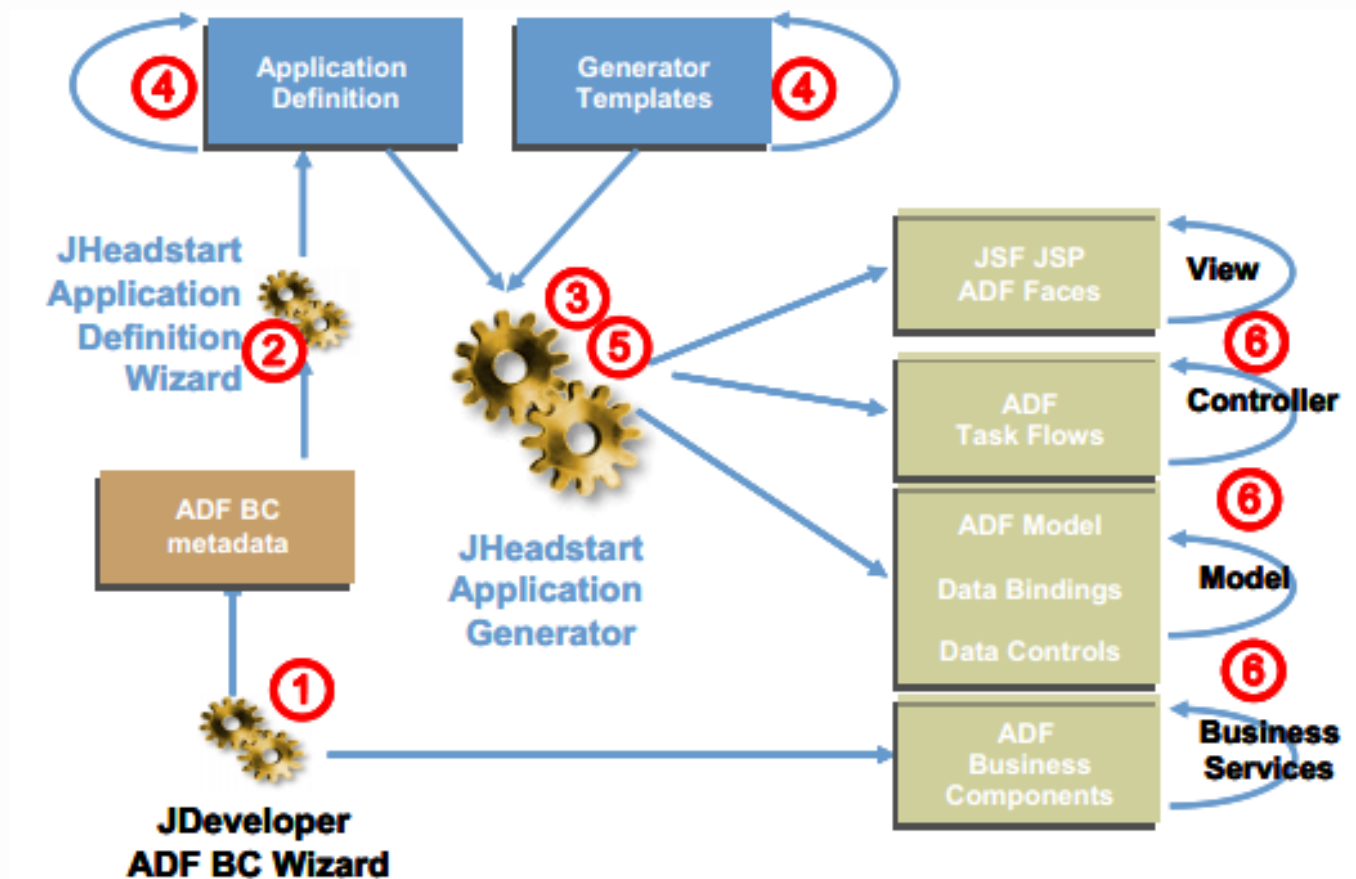


Figure 1-1 Oracle ADF Architecture

# 1 Qué es JHeadstart

Como usar JHeadstart

- Construcción a partir del metada de Designer.
- Construcción desde cero.



# 1 Qué es JHeadstart



Como usar JHeadstart

1. Creación del Business Service empleando **ADF Business Components** de JDeveloper. Este paso es independiente de JHeadstart.
2. Crear el application definition (archivo xml que representa el metadata de la definición de la aplicación a generar). Inicialmente se crea empleando el **Application Definition Wizard**, pero se puede personalizar empleando el **Application Definition Editor**.
3. General las capas Modelo (data bindings), Vista y Controlador, empleando el **Application Generator**.
4. Refinar los templates y metadata iterativamente de la generación previa y volver al paso 3.
5. Emplear las **herramientas de JDeveloper ADF** para enriquecer las páginas previamente generadas.
6. Refinar las páginas generadas iterativamente

# 1 Qué es JHeadstart

## Como usar JHeadstart



ORACLE JHeadstart Demo

Home | HRModule | Page Template | Tabbed Menu | Skin | blafplus-rich

Employees | Departments | Jobs | Locations | Countries | **Regions**

View →

- Europe
- Americas
  - Argentina
  - Brazil
  - Canada
  - Mexico
  - United States of America
    - Southlake
    - South San Francisco
    - Shipping**
    - South Brunswick
    - Seattle
- Asia
- Middle East and Africa

Edit Departments Shipping

DepartmentId: 50 ManagerId: Fripp

DepartmentName: Shipping

Employees

Action	View	Format	+	Freeze	Detach	Wrap
	EmployeeId	FirstName	LastName			
	120	Matthew	Weiss			
	121	Adam	Fripp			
	122	Pavam	Kaufing			
	123	Shanta	Vollman			
	124	Kevin	Mourgos			
	125	Julia	Nayer			
	126	Irene	Mikkilineni			
	127	James	Landry			
	128	Steven	Markle			
	129	Laura	Bissot			

Functional | Financial

Department: Shipping

ManagerId: King

JobId: ST\_MAN

PhoneNumber: 650.123.1234

Email: MWEISS

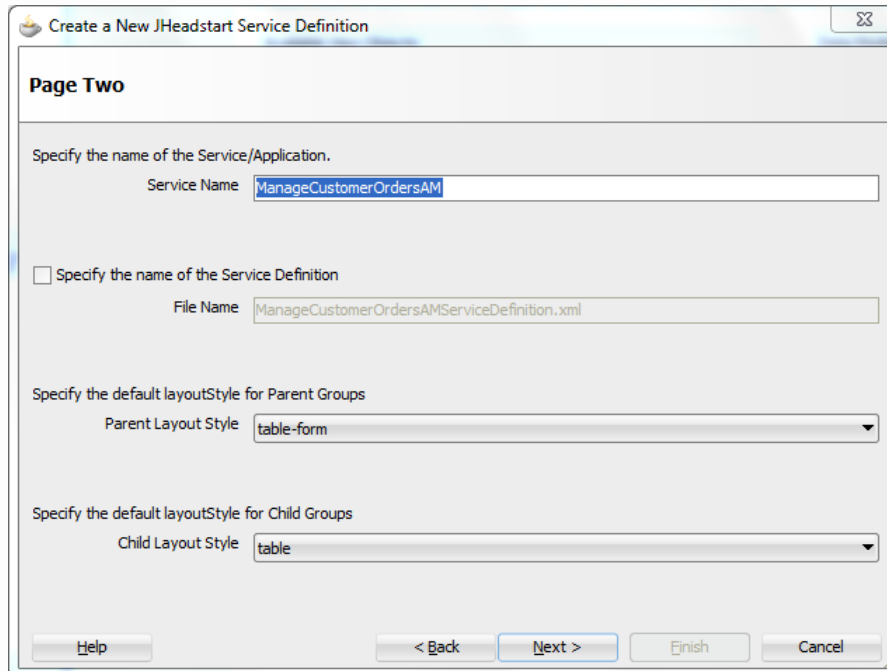
HireDate: 18-Jul-1996



# 2 Service Definition

## Fundamentos

- Es un archivo XML que define las colecciones de datos (tablas) a emplear, el estilo de presentación de las páginas (layout), la conducta de las consultas, y los componentes de interfaz de usuario a generar.
- El wizard “New Jheadstart Service Definition” crea la versión inicial del Application Definition a partir de una Application Module existente:
  - Crea un grupo (group) por cada instancia de view object.
  - Crea grupos anidados para representar las relaciones maestro-detalle.
  - Crea dominio dinámicos (dropDownList) o LOV groups, por cada view link.



The screenshot shows a wizard window titled "Create a New JHeadstart Service Definition". The window is divided into sections for specifying service and application details. The "Service Name" field is populated with "ManageCustomerOrdersAM". There is an unchecked checkbox for "Specify the name of the Service Definition", with the "File Name" field below it containing "ManageCustomerOrdersAMServiceDefinition.xml". Two dropdown menus are present: "Parent Layout Style" set to "table-form" and "Child Layout Style" set to "table". Navigation buttons at the bottom include "Help", "< Back", "Next >", "Finish", and "Cancel".

Create a New JHeadstart Service Definition

**Page Two**

Specify the name of the Service/Application.

Service Name

☐ Specify the name of the Service Definition

File Name

Specify the default layoutStyle for Parent Groups

Parent Layout Style

Specify the default layoutStyle for Child Groups

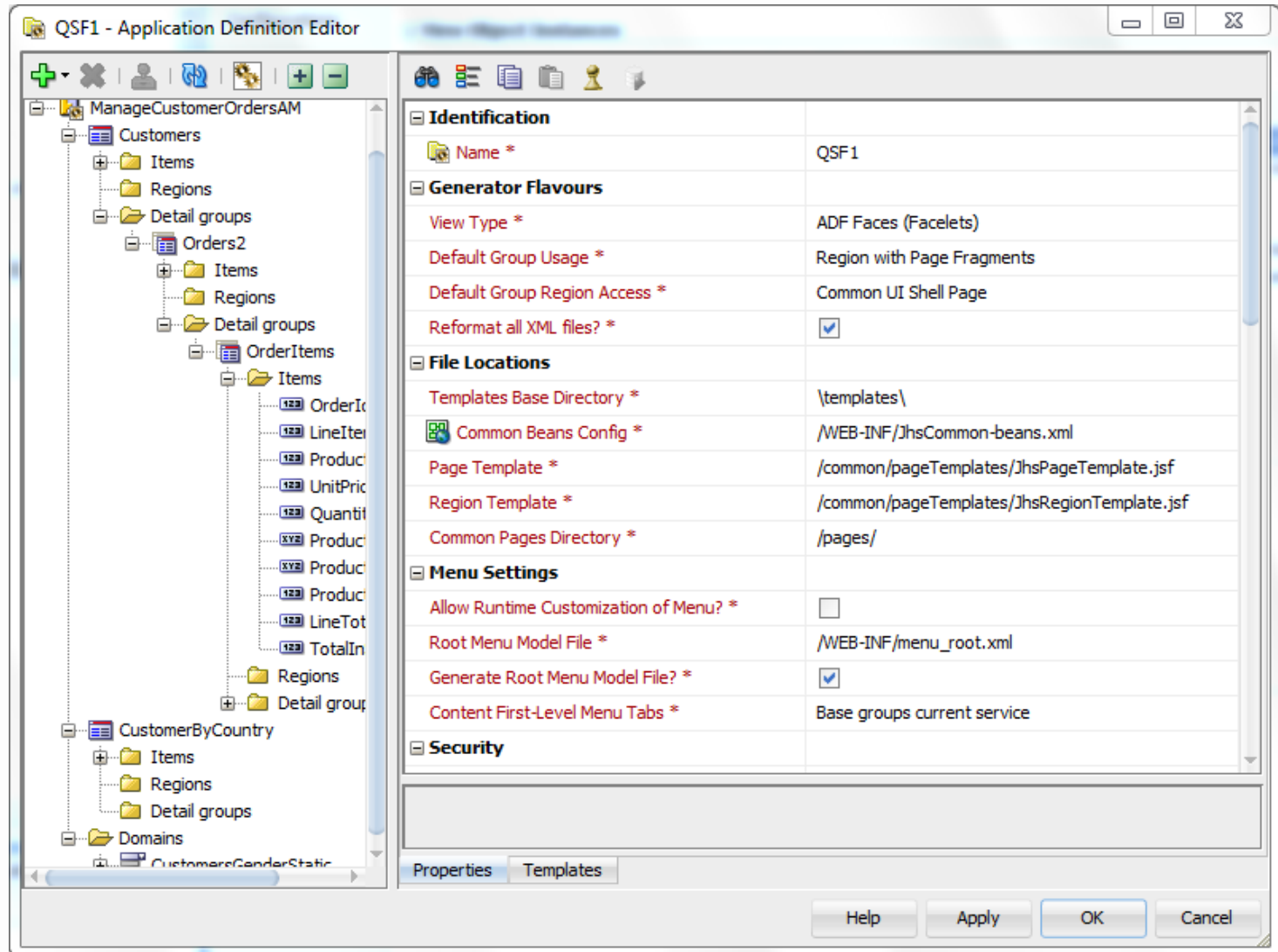
Child Layout Style

# 2 Service Definition

Editor



El Service Definition se puede editar directamente en XML o a través de Service Definition Editor.



## 2 Service Definition

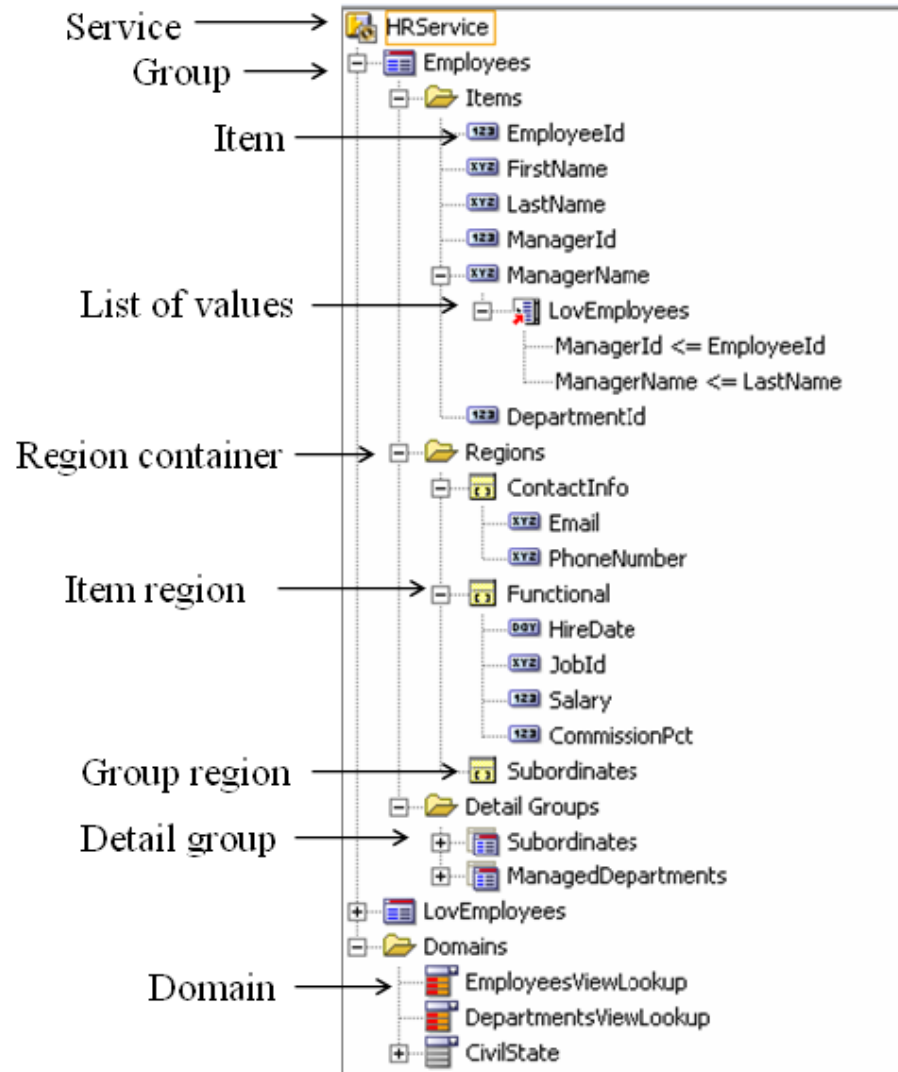
### Editor

---

- Copia y traslado de elementos usando drag-and-drop.
- Modo novato vs Modo experto.
- Copiar y pegar de múltiples propiedades a la vez.
- Despliegue del editor apropiado según la propiedad: caja de texto, checkbox, dropdown list, file browse.
- Ayuda en línea de cada una de las propiedades.

# 2 Service Definition

## Composición del Application Definition











Un Service Definition lo componen: un servicio, grupos, items, LOV, contenedores de regiones, items de región, grupos de región y dominios.

Cada tipo de elemento tiene sus propiedades respectivas.

Se presentan en una jerarquía o árbol.

# 2 Service Definition

## Composición del Application Definition - Servicio

<b>Identification</b>	
 <b>Name *</b>	ManageCustomerOrdersAM
Description	Service Definition for ManageCustomerOrdersAM
<b>Generator Flavours</b>	
Default Group Usage	
Default Group Region Access	
<b>File Locations</b>	
 <b>Service Adfc Config *</b>	/WEB-INF/adfc-config-managecustomerordersam.xml
 <b>Bounded Task Flow Template *</b>	/WEB-INF/jhs-bounded-taskflow-template.xml
 <b>Fragment Task Flow Template *</b>	/WEB-INF/jhs-fragment-taskflow-template.xml
 <b>LOV Task Flow Template *</b>	/WEB-INF/jhs-lov-taskflow-template.xml
 <b>Group Adfc Config Directory *</b>	/WEB-INF/
UI Pages Directory *	/pages/
Page Includes Directory *	/includes/
<b>Menu Settings</b>	
Menu Model File	/WEB-INF/menu-managecustomerordersam.xml
Generate Menu Model File? *	<input checked="" type="checkbox"/>
Menu Title	
<b>Java</b>	
Page Definitions Sub Package *	pageDefs
Page Lifecycle Class *	oracle.jheadstart.controller.jsf.lifecycle.JhsPageLifecycle
 <b>Data Control *</b>	ManageCustomerOrdersAMDataControl
 <b>Data Control Implementation</b>	services.ManageCustomerOrdersAM
Set Application Module Superclass to JhsApplicationModuleIm...	<input checked="" type="checkbox"/>
<b>Security</b>	
Role/Permission Prefix	
Authorized Roles/Permissions	
Insert Allowed EL Expression	
Update Allowed EL Expression	
Delete Allowed EL Expression	
<b>Runtime Customizations</b>	
Update Menu Entries? *	<input checked="" type="checkbox"/>
Allow Use of Flex Regions? *	<input type="checkbox"/>

### Servicio

Es el nodo de más alto nivel en un application definition.

Define el Application Module de BC base.

La propiedad “Overall Layout Style” determina si las páginas se accederán por medio de un menú o un wizard (Menu Style ó Wizard Style).

# 2 Service Definition

## Composición del Application Definition - Grupo

<b>Identification</b>	
Name *	Customers
Short Name	
Description	Customers
Icon	
Bound to Model Data Collection? *	<input checked="" type="checkbox"/>
Group Usage	
Group Region Access	
Add Menu Entry for this Group? *	<input checked="" type="checkbox"/>
Group Image / Icon	
<b>Group Layout</b>	
Layout Style *	table-form
Table Overflow Style	
Wizard Style Layout? *	<input type="checkbox"/>
Stack Groups on Same Page *	None
Group Width	
Group Height	
Enable Stretching? *	<input type="checkbox"/>
<b>Query Settings</b>	
Data Collection *	CustomersView1
Data Collection Implementation *	CustomersView
Data Collection Expression	
Query Bind Parameters	
Requery Condition	
<b>Search Settings</b>	
Advanced Search?	model-samePage
Advanced Search View criteria	
Advanced Search Layout Columns *	2
Quick Search?	model
Quick Search View criteria	
Single or Default Search Item	CustFirstName
Search Area Height	
Auto Query? *	<input checked="" type="checkbox"/>

### Grupo

Un servicio está conformado por uno o más grupos.

Un grupo representa un view object de BC. Es decir, que un cada grupo está atado a una colección de datos del Model.

Se definen las operaciones permitidas sobre la colección.

Un grupo puede generar más de una página, dependiendo de las propiedades “**Layout style**” (form, table, table-form, select-form, tree, tree-form, parent-shuttle, intersection-shuttle) y “**Advanced Search**” (samePage, separatePage, none).

Los grupo están anidados para conformar relaciones maestro-detalle. Los detalles se crean por medio de “detail groups”.

# 2 Service Definition

## Composición del Application Definition - Item

General	
Bound to Model Attribute? *	<input checked="" type="checkbox"/>
Name *	CustFirstName
Short Name	
Attribute Name *	CustFirstName
Value	
Java Type *	String
Display Type *	textInput
Display Settings	
Display in Form Layout? *	true
Display in Table Layout? *	true
Display in Table Overflow Area? *	false
<input checked="" type="checkbox"/> Display at Right of Item	
Display Summary Type in Table	
Prompt in Form Layout	#{HINTS\$.label}
Prompt in Table Layout	
Short prompt	
Width	#{HINTS\$.displayWidth}
Height	#{HINTS\$.displayHeight}
Maximum Length	20
Column Width	
Column Alignment	
Default Display Value	
Column Sortable? *	<input checked="" type="checkbox"/>
Column Wrap? *	<input type="checkbox"/>
Hint (Tooltip)	#{HINTS\$.tooltip}
Depends On Item(s)	
Clear/Refresh Value? *	<input type="checkbox"/>
Operations	
Update Allowed?	true
Disabled?	
Validation	
Required?	#{HINTS\$.mandatory}

### Item

Un grupo está conformado por uno o más items.

Un item representa un dato simple (columna de la tabla).

El “Display type” puede ser: text input, dropdown list, checkbox, radio group, file download.

Se puede especificar en que tipo de layout ilustrar el item.

Se pueden emplear EL para limitar la inserción o actualización.

# 2 Service Definition

## Composición del Application Definition - Regiones

- Item region: Grupo lógico de atributos con una título opcional.
- La propiedad “Layout Style” puede ser horizontalmente, Verticalmente o stacked.
- Region container: Agrupa múltiples regiones.
- Group region: Un contenedor de regiones que contiene un detail group (un detalle en la relación maestro-detalle).

HRServiceApplicationDefinition.xml - Application Definition Editor

The screenshot shows the 'Application Definition Editor' for 'HRServiceApplicationDefinition.xml'. The left pane displays a tree view of the application structure. The right pane shows the 'General' and 'Display Settings' properties for the selected 'LastName' item.

**Tree View Structure:**

- HRService
  - Employees
    - Items
      - EmployeeId
      - FirstName
      - LastName (Selected)
      - ManagerId
      - DepartmentId
    - Regions
      - ContactInfo
        - Email
        - PhoneNumber
      - Functional
        - HireDate
        - JobId
        - Salary
        - CommissionPct
      - Subordinates
    - Detail Groups
      - Subordinates
      - ManagedDepartments
    - LovEmployees
    - Domains
      - EmployeesViewLookup
      - DepartmentsViewLookup
      - CivilState
        - M (Married)
        - S (Single)

**Properties Panel:**

- General**
  - Data Bound? ☒
  - name: LastName
  - Attribute Name: LastName
  - Java Type: String
  - Display Type:
- Display Settings**
  - Display in Form Layout? true
  - Display in Table Layout? true
  - Prompt in Form Layout: LastName
  - Prompt in Table Layout:
  - Width: #{bindings.f
  - Height: #{bindings.f
  - Maximum Length: 25
  - Column Alignment:
  - Default Display Value:
  - Hint (Tooltip):
  - Depends On Item:
- Operations**

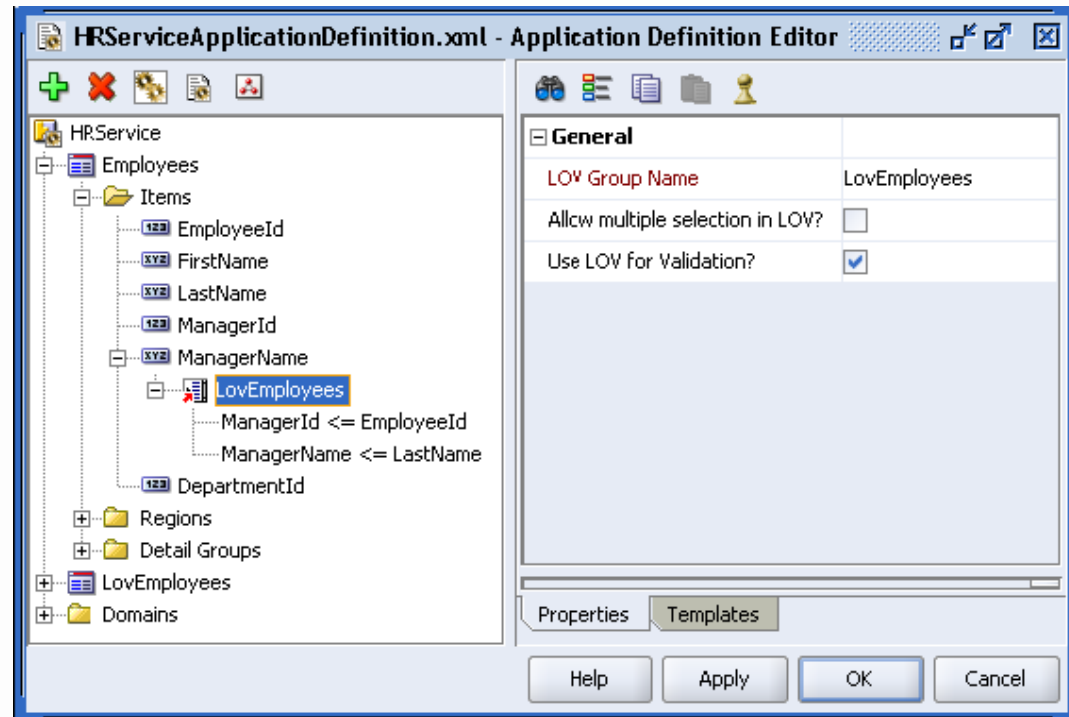
When the value of the specified item changes, this item will be re-expression is used for a property of this item that references the might be disabled or required depending on the value of another



# 2 Service Definition

## Composición del Application Definition – List of Value LOV

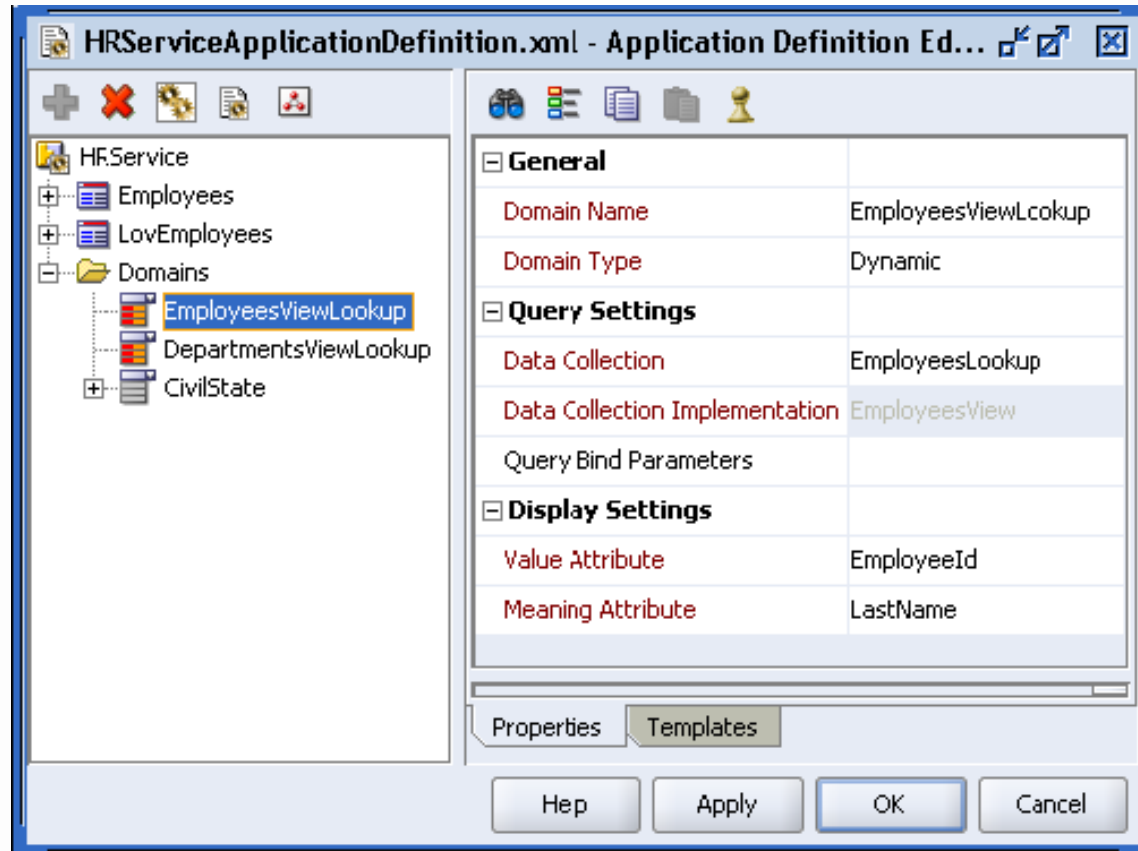
- Una LOV lista los registro de un view object en una ventana flotante.
- Una LOV está basada en un grupo con la propiedad “**Use as list of values**” en “**true**”.
- Un item puede tener asociado una LOV.
- Una LOV puede tener uno o más valores de retorno.
- La propiedad “**Use LOV for Validation**” produce una validación automática en el instante en que el usuario abandona el campo. En caso que sea necesario el campo se auto-completa.



## 2 Service Definition

### Composición del Application Definition – Dominios

- Pueden ser dominios dinámicos o estáticos.
- Los valores permitidos del dominio se definen por el Value y por el Meaning.
- Son empleados para generar radio groups, dropdown list, o checkbox.



# 3 JHeadstart Generator

## Generación

- Una vez se ha creado y personalizado el Application Definition, es posible generar la aplicación a través del JHeadstart Application Generator (JAG).
- JAG puede ser invocado directamente desde el Application Definition Editor o desde el navegador de JDeveloper.
- JAG produce:
  - Páginas JSF JSP en formato xml (.jspx)
  - El archivo faces-config.xml
  - Los archivos PageDef.
  - Los Resources Bundles para internacionalización.

# 3 JHeadstart Generator

Generación basada en templates

- El contenido de las páginas JSF y del faces-config.xml es dirigido por templates.
- El contenido dinámico está basado en VTL (Velocity Template Language).
- JHeadstart posee templates predefinidos.

```
## checkbox rendered as dropdown in search region!
<afh:script
    text="addToIgnoreChangedFields(['Search${JHS.current.item.bindingName}']);"/>

<af:selectOneChoice #ITEM_ID_IN_SEARCH() #ITEM_PROMPT_IN_SEARCH()
    #DEPENDS_ON_ITEM_PROPS_SEARCH()
    #SEARCH_ITEM_PARTIAL_TRIGGERS()
    #ITEM_HINT()
    #ITEM_VALUE_IN_SEARCH()
    #UNSELECTED_LABEL()>
    #JHS_PARSE(${JHS.current.item.domain.optionsTemplateIdentifier} ${JHS.current.model})
</af:selectOneChoice>
```

# 3 JHeadstart Generator

Preservar cambios post-generación

- No regenera las partes del sistema que han sido cambiados.
  - Apagar la generación a nivel de Servicio.
  - Apagar la generación a nivel de Grupo.
- Regenerar todo el sistema y reaplicar cambios manualmente.
- Definir los cambios en templates personalizados.

# 4 Ejecución de las aplicaciones

## Advanced Search, Editable table, Inline Overflow

**ORACLE**  
JHeadstart Demo

Home

Employees Departments Jobs Locations Countries Regions JobHistory

**Employees**

New Employee Save

☒ Result matches all conditions  
☐ Result matches any condition

Last Name contains  ID  First Name  Email   
Phone Number  Hiredate   
Job  Salary   
Commission  Manager   
Department

Find Clear Quick Search

Select Employee Details Previous 1-0 of 24 Next 10

Select	Details	*ID	First Name	*Last Name	Manager	Department	Delete?
<input checked="" type="radio"/>	Show	122	Payam	Kaufing	King	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show	123	Shanta	Vollman	King	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show	125	Julia	Nayer	Weiss	Shipping	<input type="checkbox"/>
<input type="radio"/>	Hide	127	James	Landry	Weiss	Shipping	<input type="checkbox"/>

Email  Phone Number   
Hiredate  Job   
Salary  Commission

<input type="radio"/>	Show	128	Steven	Markle	Weiss	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show	130	Mozhe	Atkinson	Fripp	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show	131	James	Marlow	Fripp	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show	133	Jason	Mallin	Kaufing	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show	136	Hazel	Philtanker	Kaufing	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show	137	Renske	Ladwig	Vollman	Shipping	<input type="checkbox"/>
<input type="radio"/>	Show						<input type="checkbox"/>

Select Employee Details Previous 1-0 of 24 Next 10

New Employee Save

Layout Style = Table-form

Advanced search = samePage

Show in Advanced Search = Seleccionado

Query Operator = setByUser

Quick Search = dropDownList

Table range Size = 10

New Rows = 1

Table overflow Style = inline

# 4 Ejecución de las aplicaciones

Advanced Search, Editable table, Inline Overflow

Filter By

Quick search  
attribute =  
Department

# 4 Ejecución de las aplicaciones

Maestro-detalle, Stacked, LOV, Shuttle

## Edit Department

<< < [4 / 27] > >>

\* ID 30  
\* Name Purchasing  
Location Seattle  
\* Manager Raphaely

Transfer Employees

Edit Employees

### Employees not in Department

King  
Kochhar  
De Haan  
Hunold  
Ernst  
Austin  
Pataballa  
Lorentz  
Greenberg  
Faviet

>  
Move

>>  
Move All

<  
Remove

<<  
Remove All

### Employees in Department

Raphaely  
Khoo  
Baida  
Tobias  
Himuro  
Colmenares

Transfer Employees

Edit Employees

Parent group:

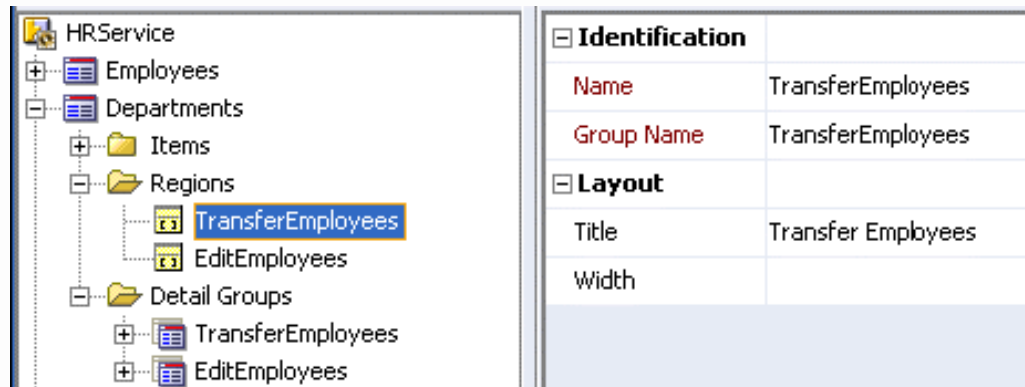
- Layout Style = table-form

Detail group:

- Layout Style = parent-shuttle
- Samepage = seleccionado

Region container:

- Layout Style = Stacked





# 4 Ejecución de las aplicaciones

## Tree, Stacked

Layout Style = tree-form

Region container:

- Layout Style = Stacked

### Employee

- Table overflow Style = right
- Samepage = seleccionado

**ORACLE**  
JHeadstart Demo

Home

Employees Departments Jobs Locations Countries **Regions** JobHistory

Regions | Countries

Search Regions >  
**Edit Department**

\* ID 60 \* Name IT  
Location 1400 Manager Hunold

New Department: Delete Department

**Employees**

Select	*ID	First Name	*Last Name
<input checked="" type="radio"/>	103	Alexander	Hunold
<input type="radio"/>	104	Bruce	Ernst
<input type="radio"/>	105	David	Austin
<input type="radio"/>	106	Valli	Pataballa
<input type="radio"/>	107	Diana	Lcrentz
<input type="radio"/>			
<input type="radio"/>			

**Functional** Contact Info

\* Hiredate 03-Jan-1990  
\* Job IT\_PROG  
Salary 9000  
Commission

**Functional** Contact Info

Save

La implementación de la seguridad (autenticación y autorización) de una aplicación ADF-JEE se puede implementar en diferentes niveles:

- Protocolo https y certificados digitales
- En la capa de presentación habilitando y deshabilitando elementos de GUI (botones, opciones de menú y campos) de acuerdo al rol del usuario.
- En BC restringiendo las operaciones del usuario
- En la base de datos restringiendo el acceso a los objetos de la base de datos. En este caso se debe emplear RLS y VPD (Row Level Security y Virtual Private Database)

# 5 Seguridad

## JAAS

JAAS (Java Authentication and Authorization Services) hace parte del estándar JEE, que garantiza el desarrollo de aplicaciones portables entre servidores de aplicaciones a través de su respectivo API:

```
FacesContext.getInstance().getExternalContext().getUserPrincipal().getName()  
FacesContext.getInstance().getExternalContext().isUserInRole("admin")
```

JAAS hace independiente la aplicación del mecanismo de almacenamiento de la información de seguridad (XML, archivos planos, tablas, LDAP)

JAZN es la implementación de JAAS en Oracle OC4J.

# 5 Seguridad

## JAAS Custom Login Module

JAAS Custom Login Module (CLM) permite almacenar la información de seguridad en cualquier repositorio.

Es muy útil en aplicaciones legacy donde el modelo de seguridad esté fuertemente acoplado con el modelo de datos del negocio y donde la información del usuario (login, password, roles, privilegios) esté almacenado en tablas de la base de datos.

A pesar que en CLM la información de los roles está en el “repositorio personalizado”, es indispensable también registrarlo en la configuración de la aplicación JEE (en el archivo web.xml)

# 5 Seguridad

## Seguridad personalizada

Si JAAS o JAAS CLM no cumplan con los exigencias de su aplicación, en JHeadStart es posible personalizar el mecanismo de autenticación y autorización completamente.

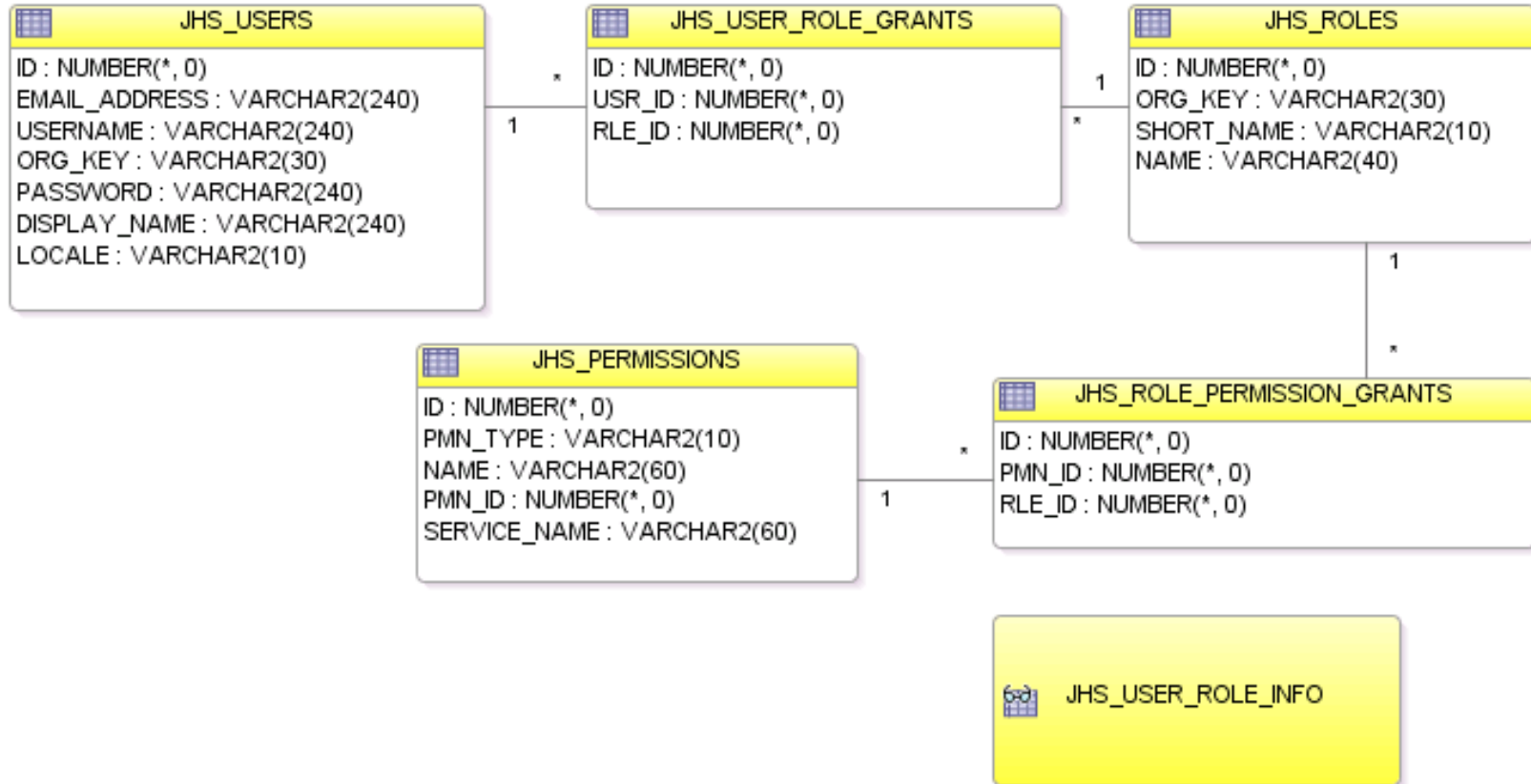
En este caso se emplea un “*servlet filter*” que redirecciona las invocaciones a la página de *login* cuando el usuario trata de acceder a una página protegida.

(`oracle.jheadstart.controller.jsf.AuthenticationFilter`)

La autenticación es implementada en el método **authenticateUser** de la clase **oracle.jheadstart.controller.jsf.bean.LoginBean** que deberá ser invocado desde la página de login personalizada

# 5 Seguridad

## Implementación de la seguridad en JHeadStart



# 5 Seguridad

## Implementación de la seguridad en JHeadStart

Las diferentes opciones de seguridad que se ofrecen en diseño, son soportadas por un modelo de tablas.

Opciones	Tablas empleadas
Authentication type = JAAS Authorization type = JAAS Authorize Using Permissions = NO	Ninguna
Authentication type = Custom	JHS_USERS
Authentication type = JAAS with CLM	JHS_USERS JHS_ROLES JHS_USER_ROLE_GRANTS JHS_USER_ROLE_INFO
Authorization type = Custom   JAAS and Custom	JHS_USERS JHS_ROLES JHS_USER_ROLE_GRANTS
Authorize Using Group Permissions = SI	JHS_ROLES JHS_ROLE_PERMISSION_GRANTS JHS_PERMISSIONS

### PERMISOS

Para evitar tener que restringir el acceso a las diferentes opciones de la aplicación a través de código alambrado, JHeadStart ofrece la definición dinámica de permisos a nivel de grupo (*group* del *application definition*) y operación (*create*, *update* y *delete*).

“Employees.Update”

La configuración dinámica de permisos permite:

- Adicionar y remover roles.
- Asignar y remover permisos a cada rol

Sin embargo, si se desea crear un nuevo permiso, será necesario modificar el código fuente de la aplicación.



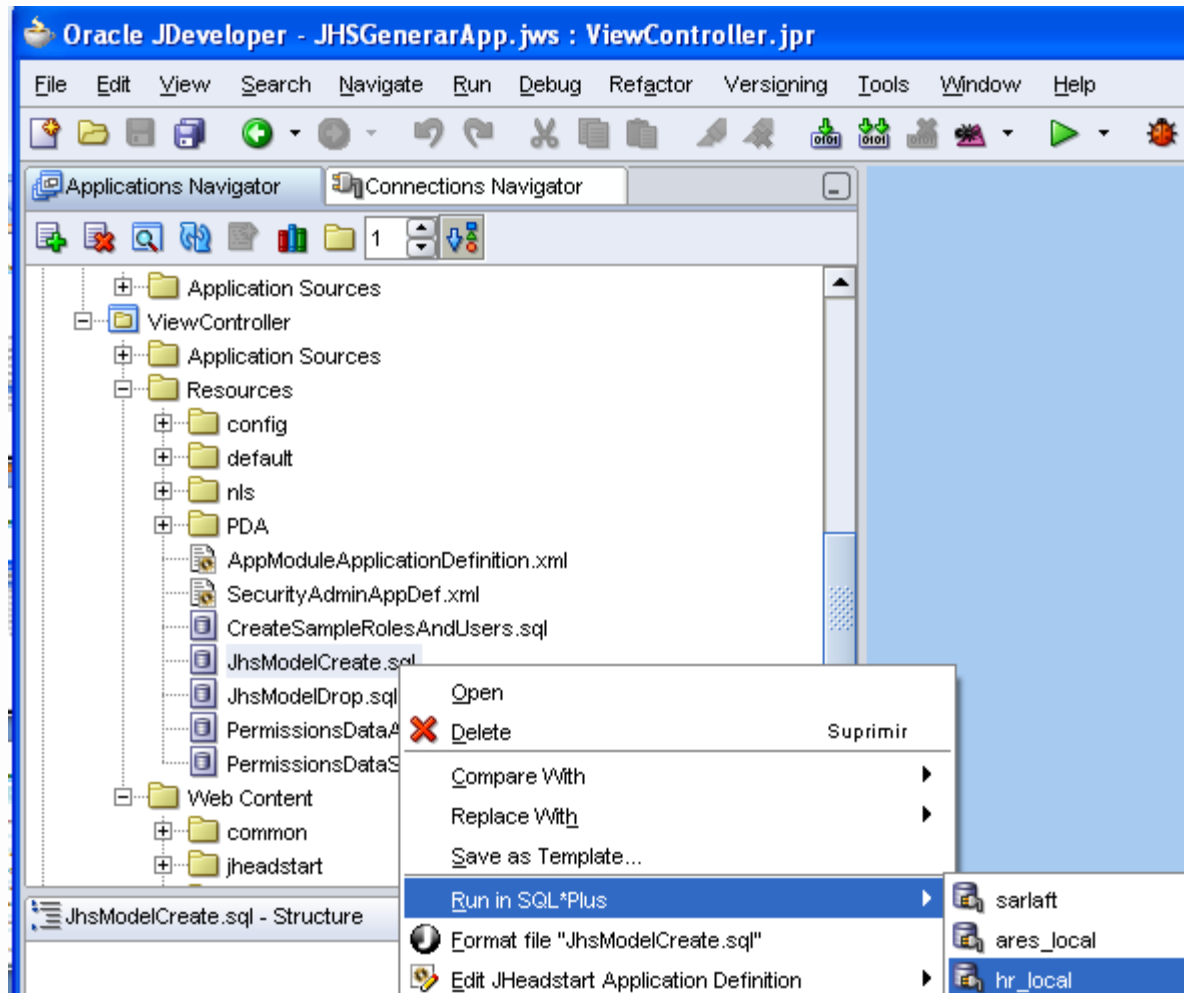
# 5 Seguridad

## Creación de las tablas de seguridad de JHeadStart

- **JhsModelCreate.sql** : Este script contiene las instrucciones de creación del modelo de tablas de JHeadStart.
  - Incluye tablas adicionales a las tablas de seguridad, por lo cual es conveniente editarlo y personalizarlo.
  - Se debe garantizar que el schema de base de datos donde residen las tablas de la aplicación puedan acceder a las tablas de seguridad.
  - Se sugiere correr este script bajo el usuario dueño de la aplicación. En caso contrario es necesario conceder privilegios y crear los sinónimos respectivos.
- El script **JhsModelCreate.sql** se encuentra dentro el proyecto **ViewController**, en la caperta **Resources**.

# 5 Seguridad

## Creación de las tablas de seguridad de JHeadStart



# 5 Seguridad

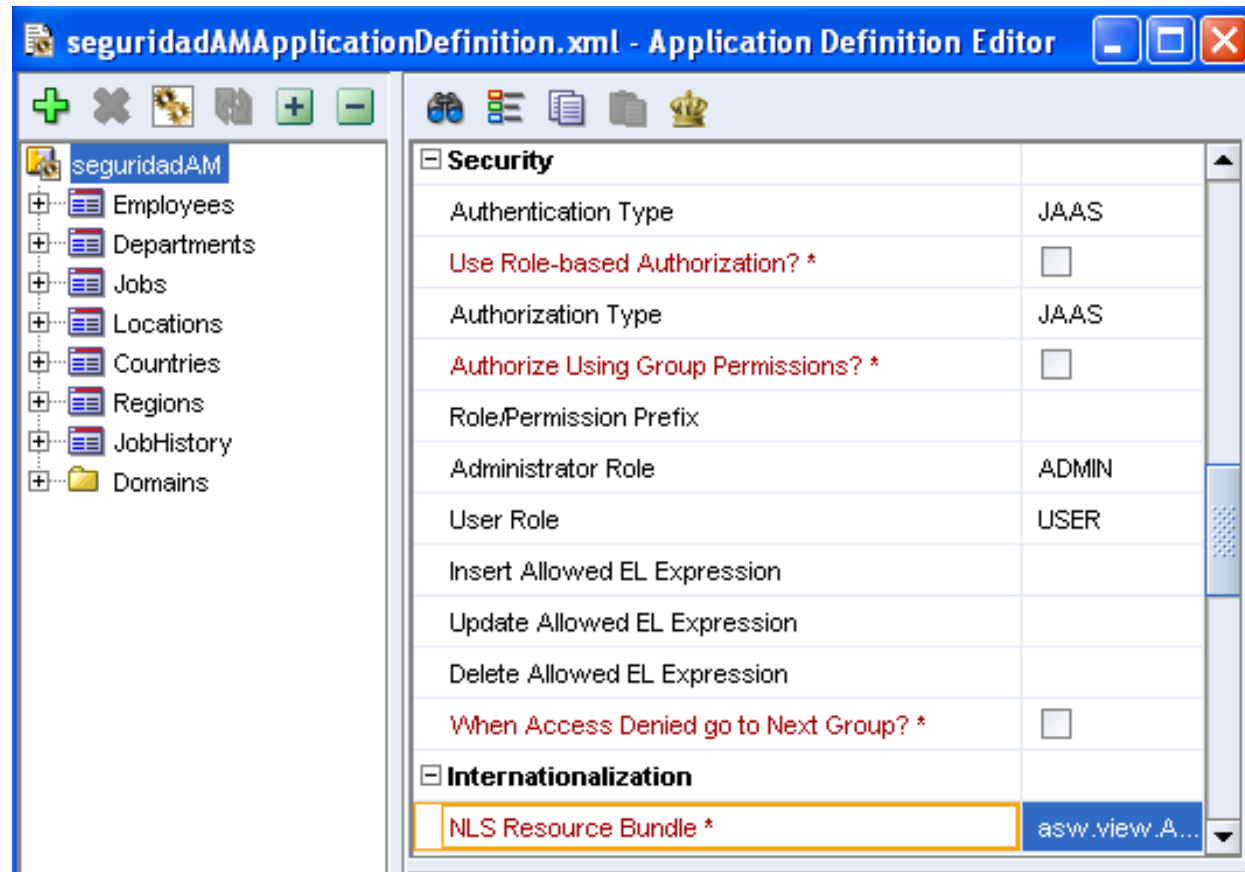
## Autenticación y autorización basada en JAAS

- No requiere de un modelo de datos de seguridad adicional.
- La seguridad es delegada al contenedor J2EE

Los roles de seguridad son especificados en el archivo web.xml

Los usuarios y roles deben ser configurados en el contenedor J2EE.

En nuestro caso (un ambiente de desarrollo) se definen en el contenedor OC4J que hace parte de JDeveloper.



# 5 Seguridad

## Autenticación y autorización basada en JAAS

### Archivos Generados

/security/pages/Login.jspx

Esta es generada a través del template loginPage.vm

Botón Logout

<login-config> en el archivo web.xml

Especifica cual es la página de login

<security-constraint> en el archivo web.xml

Se define la restricción de seguridad “AllPages”

Roles ADMIN y USER en archivo web.xml

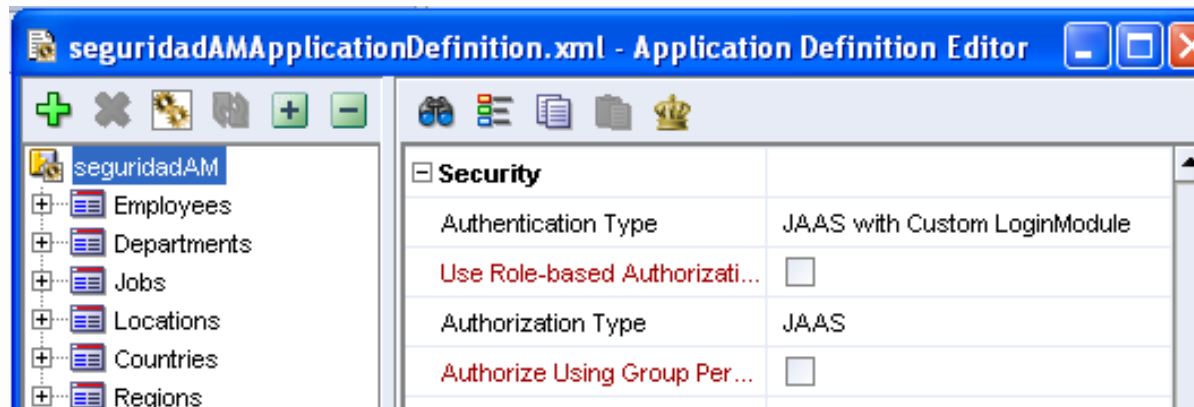
META-INF/jazn-data.xml

Especifica los usuarios SKING y AHUNOLD pertenecientes a los roles ADMIN y USER respectivamente.

# 5 Seguridad

## Autenticación usando JAAS CLM (Custom Login Module)

- Se crean registros ejemplo en las tablas JHS\_ROLES, JHS\_USERS y JHS\_USER\_ROLE\_GRANTS
- En esta caso la autenticación se realiza a través de las tablas mencionadas y no empleando la funcionalidad del contenedor JEE



# 5 Seguridad

## Autenticación usando JAAS CLM (Custom Login Module)

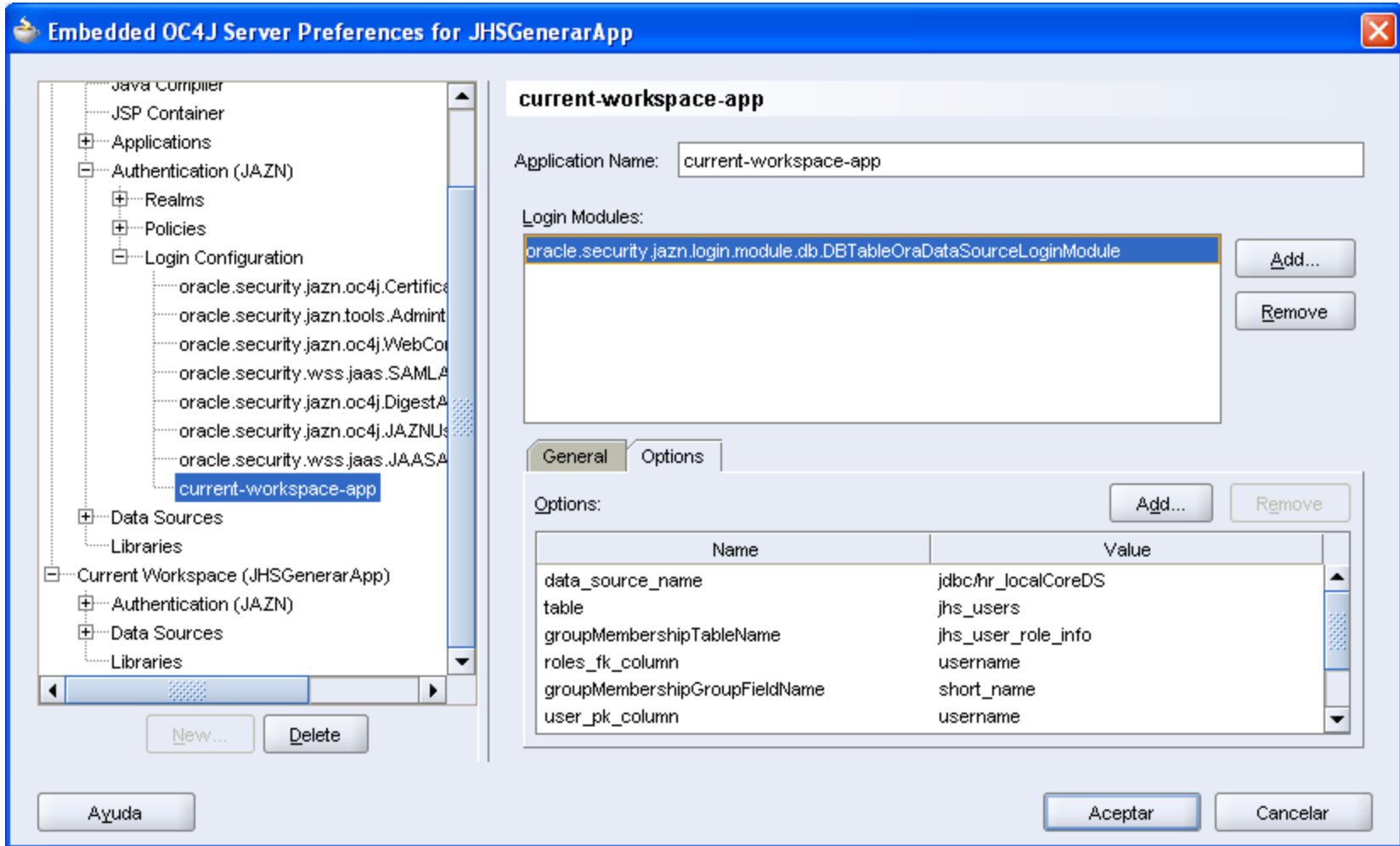
- Se deben contar con dos tablas o vistas de la base de datos. Una debe representar la entidad USUARIOS y la otra debe representar la entidad ROLES\_POR\_USUARIO.
  - La llave primaria de la tabla de usuarios debe ser el LOGIN del usuario.
  - En el caso del modelo propuesto por JHeadStart estas dos estructuras corresponden a la tabla JHS\_USERS y a la vista JHS\_USER\_ROLE\_INFO.
- En el caso de OC4J es necesario definir la aplicación “current-workspace-app” en el archivo system-jazn-data.xml, basada en la clase oracle.security.jazn.login.module.db.DBTableOraDataSourceLoginModule.
  - Con la definición de las siguientes opciones:

data_source_name	jdbc/hr_localCoreDS
table	jhs_users
groupMembershipTableName	jhs_user_role_info
roles_fk_column	username
groupMembershipGroupFieldName	short_name
user_pk_column	username
passwordField	password
usernameField	username
casing	sensitive

# 5 Seguridad

## Autenticación usando JAAS CLM (Custom Login Module)

- La configuración del archivo system-jazn-data.xml se puede realizar accediendo en Jdeveloper a la opción Tool -> Embedded OC4J Server



# 5 Seguridad

## Autorización basada en roles o en permisos

- En caso que se haber empleado la opción **Authorization Type**, es posible contar con dos formas de autorizar el acceso a las diferentes opciones de la aplicación:

- Basada en roles

☐ **Authorization**

Authorized Roles/Permissions	ADMIN, HR_MANAGER
------------------------------	-------------------

- Se realiza a nivel de cada grupo
- Se restringe a nivel de presentación y a nivel del controlador
- No requiere tablas de configuración

- Basada en permisos

☐ **Security**

Authentication Type	JAAS
Use Role-based Authorization? *	<input checked="" type="checkbox"/>
Authorization Type	JAAS
Authorize Using Group Permissions? *	<input checked="" type="checkbox"/>

- El nombre del permiso corresponde con el nombre del grupo
- No es necesario especificar la propiedad **Authorized Roles/Permissions** en cada grupo
- Requiere la configuración en las tablas: JHS\_PERMISSIONS, JHS\_ROLES, JHS\_ROLE\_PERMISSION\_GRANTS



# 5 Seguridad

## Autorización basada en roles o en permisos

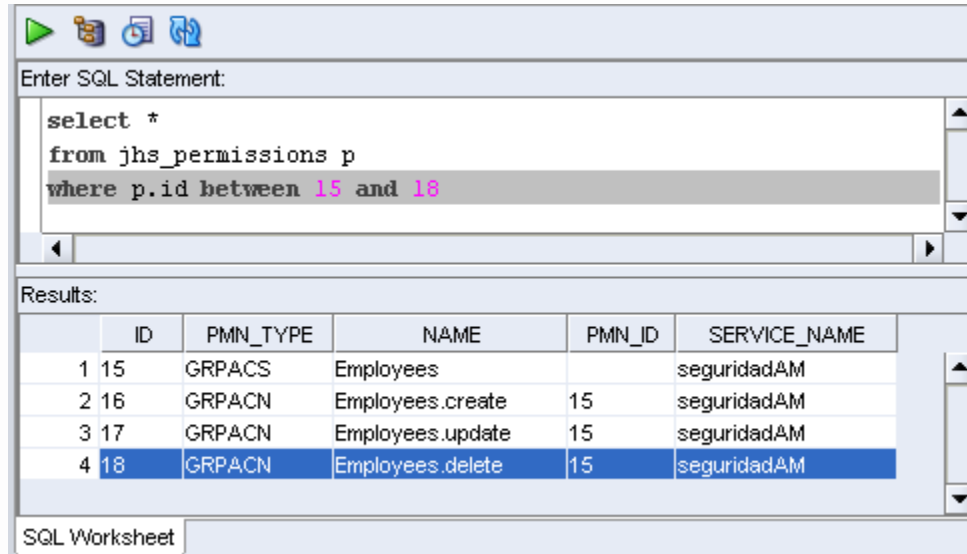
- La expresión EL `#{jhsUserRoles['ROL_ADMIN', 'ROL_USER']}` evalúa a verdadero en caso que el usuario actual pertenezca a alguno de los roles listados.
- RESTRINGIR LAS OPERACIONES DE Create, Update y Delete

☐ <b>Security</b>	
Authentication Type	JAAS
Use Role-based Authorization? *	<input checked="" type="checkbox"/>
Authorization Type	JAAS
Authorize Using Group Permissions? *	<input checked="" type="checkbox"/>
Role/Permission Prefix	
Administrator Role	ADMIN
User Role	USER
Insert Allowed EL Expression	<code>#{jhsUserRoles['\$GROUP_NAME\$.create']}</code>
Update Allowed EL Expression	<code>#{jhsUserRoles['\$GROUP_NAME\$.update']}</code>
Delete Allowed EL Expression	<code>#{jhsUserRoles['\$GROUP_NAME\$.delete']}</code>
When Access Denied go to Next Group? *	<input checked="" type="checkbox"/>

# 5 Seguridad

Autorización basada en permisos

## Configuración de permisos a través de las tablas JHS



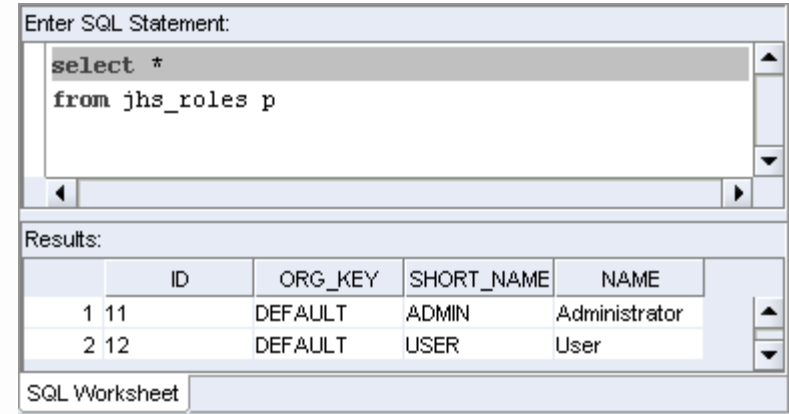
Enter SQL Statement:

```
select *  
from jhs_permissions p  
where p.id between 15 and 18
```

Results:

	ID	PMN_TYPE	NAME	PMN_ID	SERVICE_NAME
1	15	GRPACS	Employees		seguridadAM
2	16	GRPACN	Employees.create	15	seguridadAM
3	17	GRPACN	Employees.update	15	seguridadAM
4	18	GRPACN	Employees.delete	15	seguridadAM

SQL Worksheet



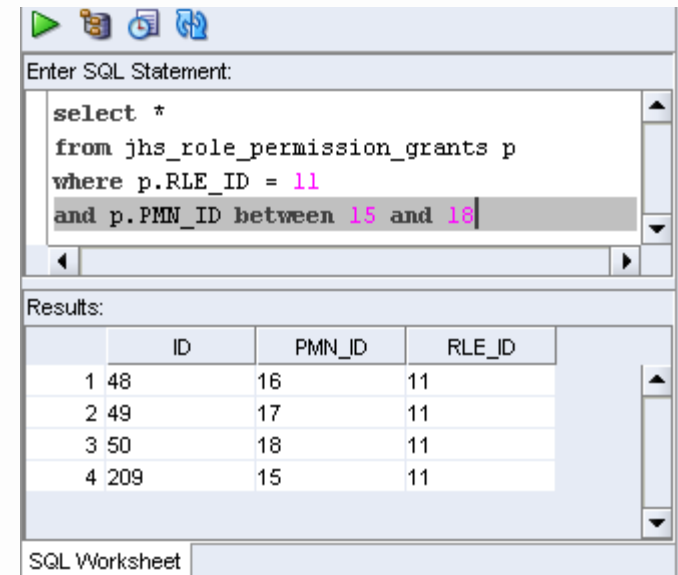
Enter SQL Statement:

```
select *  
from jhs_roles p
```

Results:

	ID	ORG_KEY	SHORT_NAME	NAME
1	11	DEFAULT	ADMIN	Administrator
2	12	DEFAULT	USER	User

SQL Worksheet



Enter SQL Statement:

```
select *  
from jhs_role_permission_grants p  
where p.RLE_ID = 11  
and p.PMN_ID between 15 and 18
```

Results:

	ID	PMN_ID	RLE_ID
1	48	16	11
2	49	17	11
3	50	18	11
4	209	15	11

SQL Worksheet

# 6 Menús

## Menú estático

En cuanto al menú generado por JHeadStart tenemos:

- Es generado por los templates pageMenu.vm, menu1.vm y menu2.vm (localizados en default/misc/menu).
- La estructura del menú es generada en la region jspix denominada <nombreApplicationModule>Menu1Tabs.jspix

## Personalización a un menú tipo menuList.

### pageMenu.vm

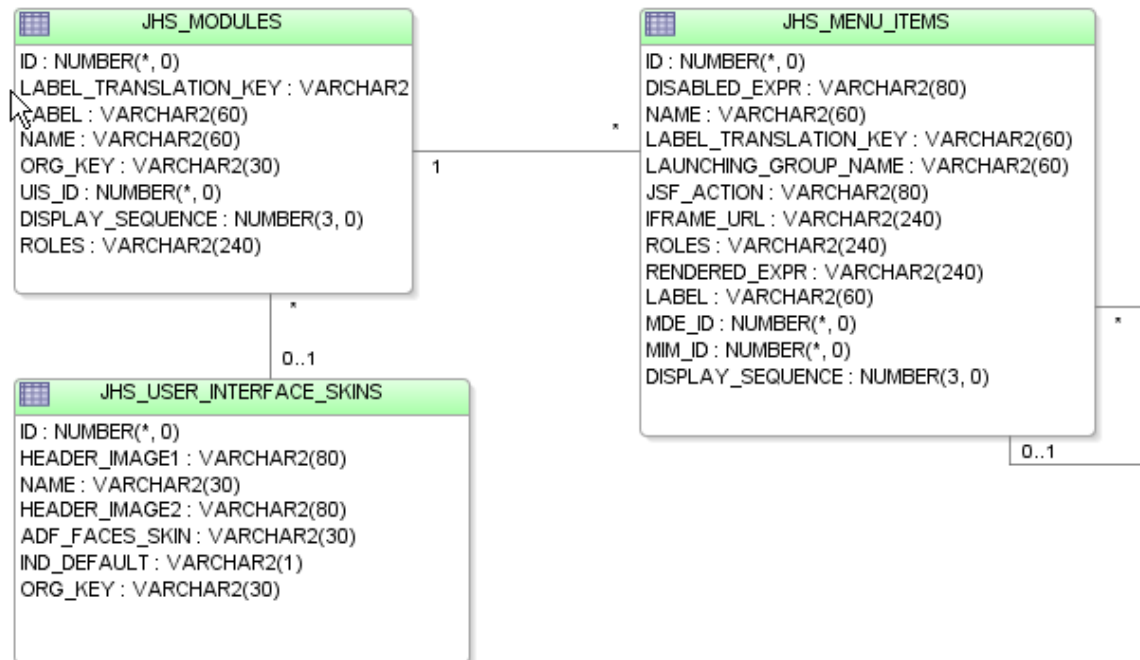
```
<f:facet name="menu3">
  #set ($includeName = "${JHS.service.name}Menu1Tabs")
  #INCLUDE_OPEN($includeName ${JHS.menuGenerator.menu1})
  #INCLUDE_ATTRIBUTE($includeName "selectedTab" ${JHS.cu
  #INCLUDE_CLOSE($includeName)
</f:facet>
```

### menu1.vm

```
<af:menuList>
#foreach ($menuItem in $JHS.current.menu.menuItems)
  <af:commandMenuItem text="${JHS.nls($menuItem.page.
    action="${JHS.facesConfigGenera
    selected="#{attrs.selectedTab==
    #if ($menuItem.page.group.roles
      rendered="#{jhsUserRoles['$me
    #end
  >
  <f:actionListener type="oracle.jheadstart.control
  <f:actionListener type="oracle.jheadstart.control
  <af:resetActionListener/>
</af:commandMenuItem>
#end
</af:menuList>
```

# 6 Menús

## Menú dinámico



- Habilitar la opción “Allow Runtime Customization of Menu?”
- `DynamicMenuData<applicationModule>.sql`
- `MenuAdmidAppDef.xml`

# 7 Manejo de mensajes de error

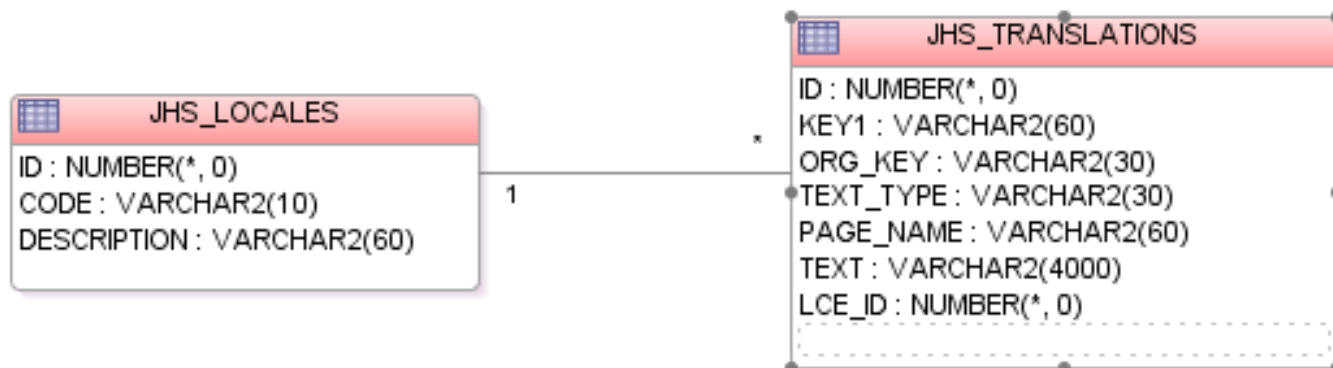
Mensajes de error a través de la base de datos

TallerFlexAMApplicationDefinition.xml - Application Definition Editor

The screenshot shows the 'TallerFlex.AM' project in the 'Application Definition Editor'. The left pane displays a tree view with folders: Employees, Departments, Items, Regions, RegionDepartamentosFlexible, Detail Groups, and Domains. The right pane shows the 'Internationalization' settings for the application.

Internationalization	
NLS Resource Bundle *	asw.view.ApplicationResources
Resource Bundle Type *	databaseTable
Override NLS Resource Bundle Entries? *	<input checked="" type="checkbox"/>
Generate NLS-enabled prompts and tabs? *	<input checked="" type="checkbox"/>
Generator Default Locale *	es
Generator Locales	nl
Read User Locale From *	Browser Setting
Generate Locale Switcher? *	<input type="checkbox"/>

Below the Internationalization section, the 'Runtime Customizations' section is visible but empty.



# 7 Manejo de mensajes de error

## Manejo de errores

- Existen diferentes comportamientos en los mensajes de error que puede ser necesario modificar:
  - Existen mensajes de error que se derivan de otros errores previos. Estos mensajes puede confundir al usuario.
  - Los mensajes de error ORA-20xxx muchas veces no son significativos para el usuario.
  - Los mensajes generados a través de JboException(“mensaje”) pierden la capacidad de internacionalización.
- El manejo de errores reside principalmente en la clase ErrorReportingUtils y en la clase JboException.

# 7 Manejo de mensajes de error

## Manejo de errores

- Crear una clase derivada de ErrorReportingUtils.

```
public class HRErrorReportingUtils extends ErrorReportingUtils {  
    public HRErrorReportingUtils() {  
    }  
    public String getOra20xxx(SQLException sqlException) {  
        return super.getOra20xxx(sqlException);  
    }  
    public String getMessage(Exception ex, Locale locale) {  
        return super.getMessage(ex, locale);  
    }  
    public void reportErrors(PageLifecycleContext ctx) {  
        super.reportErrors(ctx);  
    }  
}
```

- Cambiar en JhsCommon-beans.xml la clase que representa el managed-bean errorReportingUtils por la nueva clase. (para preservar los cambios modificar el template default/misc/facesConfig/jhsCommonBeans.vm)

**<managed-bean>**

**<managed-bean-name>**errorReportingUtils**</managed-bean-name>**

**<managed-bean-class>**HR.util.HRErrorReportingUtils**</managed-bean-class>**

**<managed-bean-scope>**application**</managed-bean-scope>**

**<managed-property>**

# 7 Manejo de mensajes de error

## Manejo de errores

- Crear una clase derivada de JboException.

```
public class HRException extends JboException {  
    public HRException(String errorCode, Object[] params) {  
        super(ApplicationResources.class, errorCode, params);  
    }  
    public HRException(String productAndErrorCode) {  
        super(ApplicationResources.class, productAndErrorCode, null);  
    }  
    public String getProductCode() {  
        return "";  
    }  
}
```

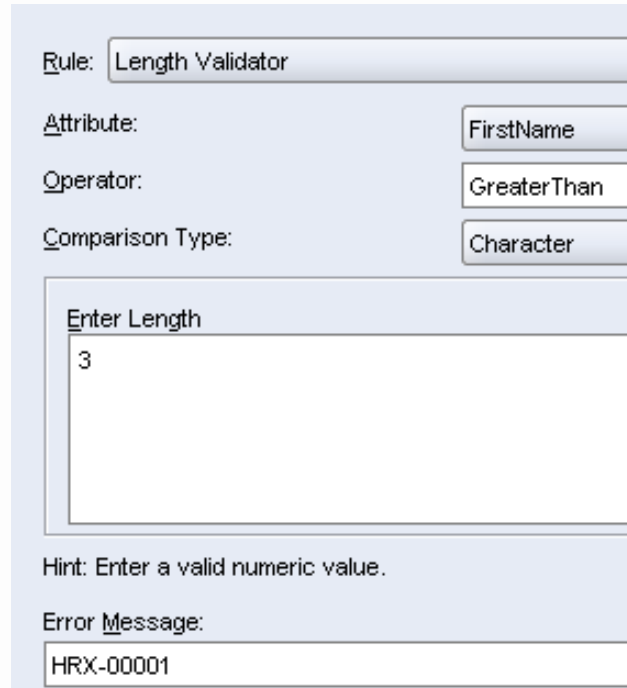
- ApplicationResources.class corresponde a una clase creada dentro del proyecto que especifica la localización especificada por el desarrollador.
- getProductCode retorna la cadena vacía ya que el código del producto hace parte del parámetro productAndErrorCode.



# 7 Manejo de mensajes de error

## Manejo de errores – personalizar excepciones de BC

- Crear una validación que retorne como mensaje un código personalizado de la forma HRX-xxxxx



Rule: Length Validator

Attribute:

Operator:

Comparison Type:

Enter Length

Hint: Enter a valid numeric value.

Error Message:

- Insertar en la tabla JHS\_TRANSLATIONS el mensaje HRX-00001

ID	KEY1	ORG_KEY	TEXT_TYPE	PAGE_NAME	TEXT	LCE_ID
800	HRX-00001	DEFAULT			Nombre muy corto	545
801	HRX-00001	DEFAULT			Name too short	341

# 7 Manejo de mensajes de error

## Manejo de errores – personalizar excepciones de BC (Cont)

- Sobrecribir el método getMessage en la clase HRErrorReportingUtils para dar manejo a las excepciones cuyo mensaje inicie con HRX-. Por omisión esta clase solo maneja aquellas mensajes JBO-.
- Hacer que los mensajes HRX soporte internacionalización.

```
public String getMessage(Exception ex, Locale locale) {  
    String message = ex.getMessage();  
    if (message.startsWith("HRX-")) {  
        String productAndErrorCode = message.substring(0,9);  
        HRException mcex = new HRException(productAndErrorCode);  
        // mensaje en formato {productCode}-{errorCode}: {messageBody}  
        message = super.getMessage(mcex, locale);  
        //Remover "-" inicial  
        if (message.startsWith("-")) message = message.substring(1);  
        return message;  
    }  
    else {  
        return super.getMessage(ex, locale);  
    }  
}
```

# 7 Manejo de mensajes de error

## Manejo de errores – personalizar ORA-20XXX

- Sobrecribir el método `getOra20xxx` en la clase `HRErrorReportingUtils` para dar manejo a las excepciones `SQLException`.

```
public String getOra20xxx(SQLException sqlException) {  
    //obtener el locate (internacionalizacion)  
    FacesContext fc = FacesContext.getCurrentInstance();  
    ValueBinding vb = fc.getApplication().createValueBinding("#{data}");  
    BindingContext bc = (BindingContext) vb.getValue(fc);  
    Locale userLocale = bc.getLocaleContext().getLocale();  
  
    String sqlMessage = this.getMessage(sqlException, userLocale);  
    return sqlMessage;  
}
```


- Modificar el método `getMessage` para incluir los mensajes que inicie con “ORA-20”.

```
public String getMessage(Exception ex, Locale locale) {  
    String message = ex.getMessage();  
    if (message.startsWith("HRX-") || message.startsWith("ORA-20")) {  
        String productAndErrorCode = message.substring(0, 9);  
        ...  
    }  
}
```

# 7 Manejo de mensajes de error

## Manejo de errores – personalizar ORA-20XXX (Cont.)

- Insertar en la tabla JHS\_TRANSLATIONS el mensaje ORA-20001

	ID	KEY1	ORG_KEY	TEXT_TYPE	PAGE_NAME	TEXT	LCE_ID
▶	802	ORA-20001	DEFAULT			Error en el correo	545
	803	ORA-20001	DEFAULT			Email error	341

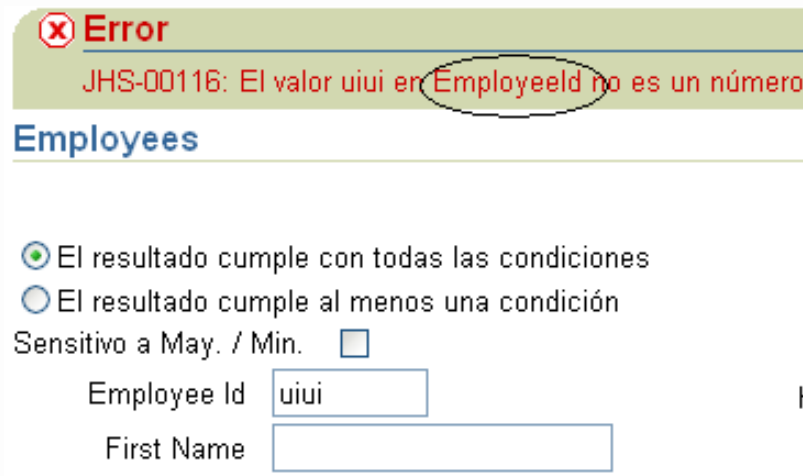
- Generar la excepción a nivel de la base de datos para realizar la prueba

```
CREATE OR REPLACE TRIGGER TRG_PRUEBA
AFTER UPDATE OF EMAIL ON EMPLOYEES
BEGIN
    raise_application_error( -20001, 'Error de BD' );
END;
```

# 7 Manejo de mensajes de error

## Manejo de errores – personalizar errores

- Al emplear el formulario de búsqueda, es posible introducir valores alfabéticos donde se esperan valores numéricos. JHeadStart captura el error y lo muestra al usuario. La deficiencia es que ilustra el nombre del atributo del ViewObject donde ocurrió el error. Este nombre de atributo no es significativo para el usuario.
- Como se corrige dicho “bug”



The screenshot shows a web application interface. At the top, there is a green error banner with a red 'x' icon and the text 'Error'. Below this, a red message reads 'JHS-00116: El valor uiui en EmployeeId no es un número'. The word 'EmployeeId' is circled in red. Below the error message, the word 'Employees' is displayed in blue. Underneath, there are two radio buttons: the first is selected and labeled 'El resultado cumple con todas las condiciones', and the second is unselected and labeled 'El resultado cumple al menos una condición'. Below these is a checkbox labeled 'Sensitivo a May. / Min.' which is also unselected. At the bottom, there is a search form with two input fields: 'Employee Id' containing the text 'uiui' and 'First Name' which is empty. To the right of the 'Employee Id' field is a small 'H' icon.