

Web Application Security

Implementing the Superstition in JDeveloper

Duncan Mills

Senior Director of
Product Management,
Application Development Tools

Peter Koletzke

Technical Director &
Principal Instructor



ORACLE®

quovera

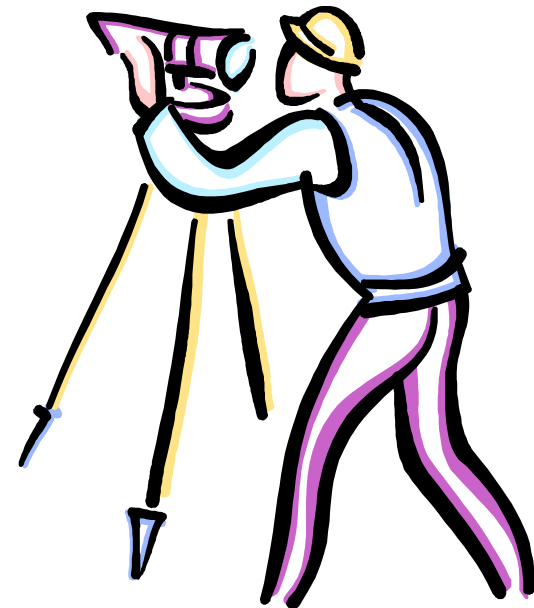
Believe It or Not

Security is mostly a superstition.
It does not exist in nature,
nor do the children of men
as a whole experience it.
Avoiding danger is no safer
in the long run than outright exposure.
Life is either a daring adventure
or nothing.

—Helen Keller (1880-1968)

Survey

- Job responsibilities?
 - DBA, developer
- Languages?
 - PL/SQL
 - Java – JSF
 - C++
 - Other
- Tools?
 - Developer Forms/Reports
 - JDeveloper
 - Other



Agenda

- Overview
- Application security tasks
- UI techniques

Slides and white paper are on your conference CD.

The white paper contains a hands-on practice

Coming Up

Tues 3:30

Rapid Development for Mobile Devices with JDev and ADF, Duncan

Tues 4:45

... and UX Begat ADF Faces: How Rich is ADF Faces Rich Client?, Peter

Wed 9:45

Deploying Applications to WebLogic Server, Duncan and Peter



Application Security Objectives

- Ultimate security may just be superstition, however, data must be protected
 - Need to make breaking in as difficult as possible
- Web apps are more accessible to hackers
- Protections needed for
 - Application access
 - Application functions (no SQL injection, cross-site scripting)
 - Data access
 - Data visibility
 - Tracking user activity

Assumes the server and file systems are protected.



Two Primary Operations

- Authentication
 - Validate that the user is who she/he claims to be
 - Normally done with passwords
 - With extra equipment, could be something else
 - Retinal scan, thumbprint, biometric scanners? DNA?
- Authorization



How to Implement the Superstition

- Use recognized, prebuilt, proven, supported security technologies
- Java Authentication and Authorization Services (JAAS)
 - Java API library in the Java SE Development Kit (JDK or J2SDK))
 - Accessible through Oracle Platform Security Services (OPSS) – a service of WebLogic Server
- Oracle ADF Security
 - Built on top of OPSS
 - Uses standard ADF declarative techniques
 - Once you turn it on, you need to define access for all pages in the application
 - Note: **not** www.adfsecurity.com



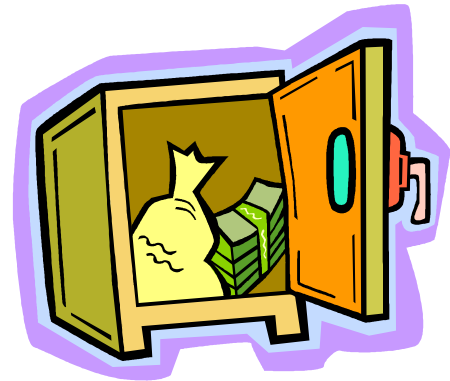
The Security Policy

- A definition of privileges in ADF Security
 - Contained in a *Security Policy Store*
- Create one or more in an application
- Principals
 - One or more roles (groups of users) who are granted access
- Resources
 - Bounded task flow – including all flows under it
 - Web pages that use ADF bindings
 - Entity objects and entity object attributes
- Permissions
 - Privileges such as View, Customize, Grant, Edit, Personalize



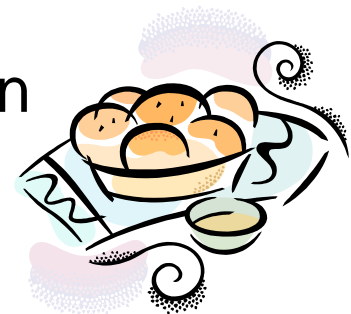
The User Repository

- The storehouse of user and enterprise role information
 - A.k.a., *credentials store* or *identity store*
- OPSS can tap two types of user repositories
 - XML (Extensible Markup Language) properties file
 - LDAP (Lightweight Directory Access Protocol)
 - A communications protocol
 - Oracle Internet Directory (OID)
 - Used for Single Sign-On (SSO)
 - OID can read other LDAP providers
 - E.g., Microsoft Active Directory, WLS LDAP

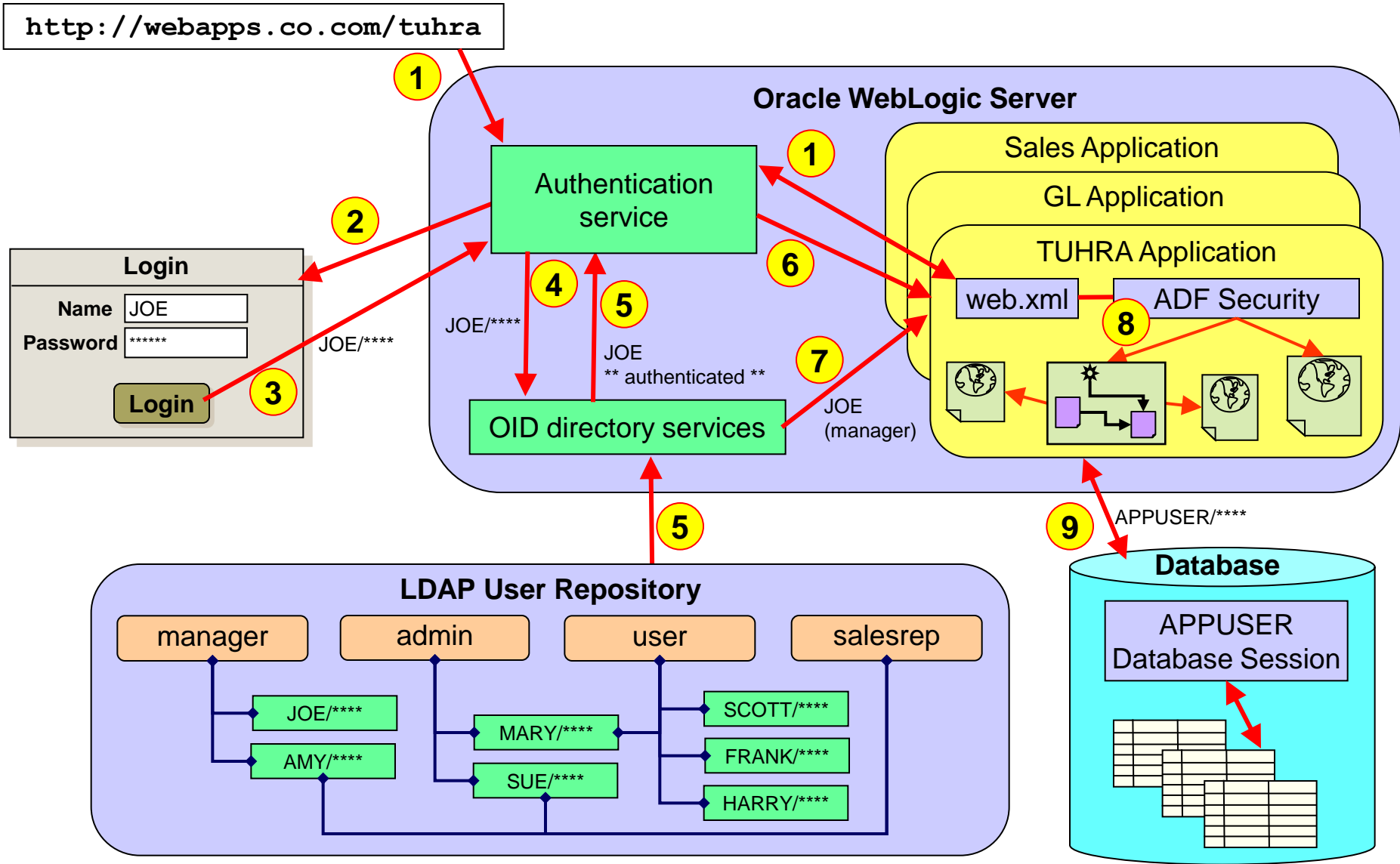


What's a Role?

- Users have a “role” or within the enterprise
 - AKA “Enterprise Roles”
 - “Warehouse Clerk”, “HR Manager”, “Chief Bottlewasher”
 - A single user will usually have multiple roles
 - Totally dependant on the business organization
 - May change over time for a single user
- Applications also have the concept of “role”
 - **Not** the same thing
 - Application roles define functional areas within the application’s “responsibilities”
 - “approver”, “page manager”, “user” etc...

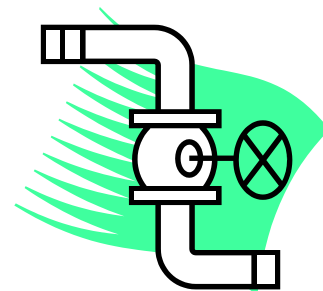


Application Security Flow



Application Security Flow

1. User sends HTTP request including a context root indicating a particular application. The request determines that ADF security is active in the application.
2. The authentication service determines the method (XML or LDAP) and presents a login page.
3. The user enters an ID and password and submits the login page.
4. The authentication service requests OID to verify the user and password.
5. OID verifies the password from the LDAP source and indicates pass or fail to the authentication service.
6. The authentication service accesses the application and places the user name into the HTTP session state.
7. The application can request the username or group (role, in this example, “manager”) to which the user belongs.
8. web.xml activates ADF Security for authorization to specific resources like pages and task flows.
9. The application connects to the database using the application database user account (APPUSER) written into a configuration file.



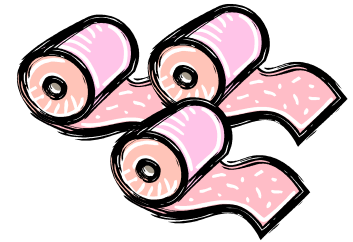
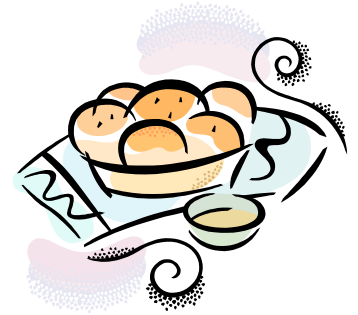
Agenda

- Overview
- Application security tasks
- UI techniques



Check List

- Enable ADF Security
- Create application roles
- Define a credentials store
- Map application roles to credential store “enterprise” roles
- Implement authorization to create security policies



Enable ADF Security

Configure ADF Security - Step 1 of 5

Enable ADF Security

Select the ADF security model you want to enable. If you just want to configure Java EE security without any ADF features, see [Developing Secure Applications](#) for help.

Security Model:

- ADF Authentication and Authorization**
Java EE security extended to support [ADF authentication and authorization](#). This is recommended if you're building an ADF web application, including WebCenter.
- ADF Authentication**
Java EE security extended to support only ADF authentication.
- Remove ADF Security Configuration**
Remove all metadata previously generated by this wizard to enable ADF Security.

ADF Security

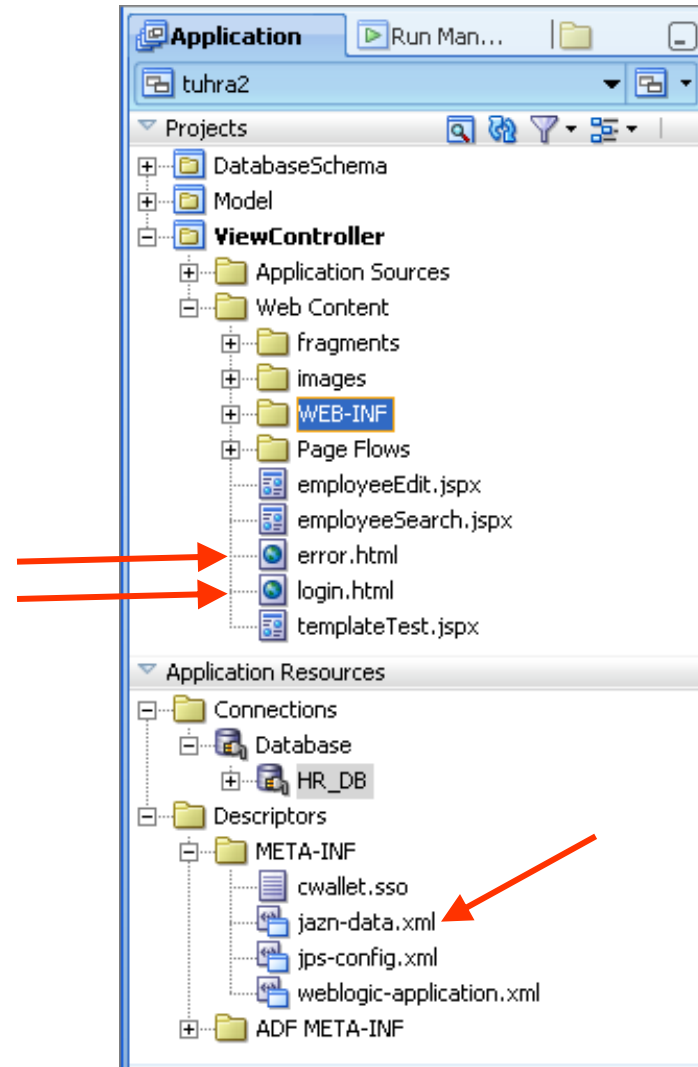
- [Authentication Type](#)
- Automatic Policy Grants
- Authenticated Welcome
- Summary

Help < Back **Next >** Finish Cancel

- Application | Secure | Configure ADF Security

Configure ADF Security Wizard

- Form-based authentication
 - Custom pages for login and error
- No automatic grants
- Redirect upon successful authentication
- Creates
 - login.html
 - error.html
 - jazn-data.xml
- Updates
 - web.xml (auth type and page names)
 - weblogic.xml
 - Look at it for security-role-assignment
 - Maps principals (users) to roles



web.xml Entries - 1

```
<servlet-mapping>
  <servlet-name>adfAuthentication</servlet-name>
  <url-pattern>/adfAuthentication</url-pattern>
</servlet-mapping>
<jsp-config>
  <jsp-property-group>
    <url-pattern>*.jsff</url-pattern>
    <is-xml>>true</is-xml>
  </jsp-property-group>
</jsp-config>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>adfAuthentication
    </web-resource-name>
    <url-pattern>/adfAuthentication</url-pattern>
  </web-resource-collection>
```

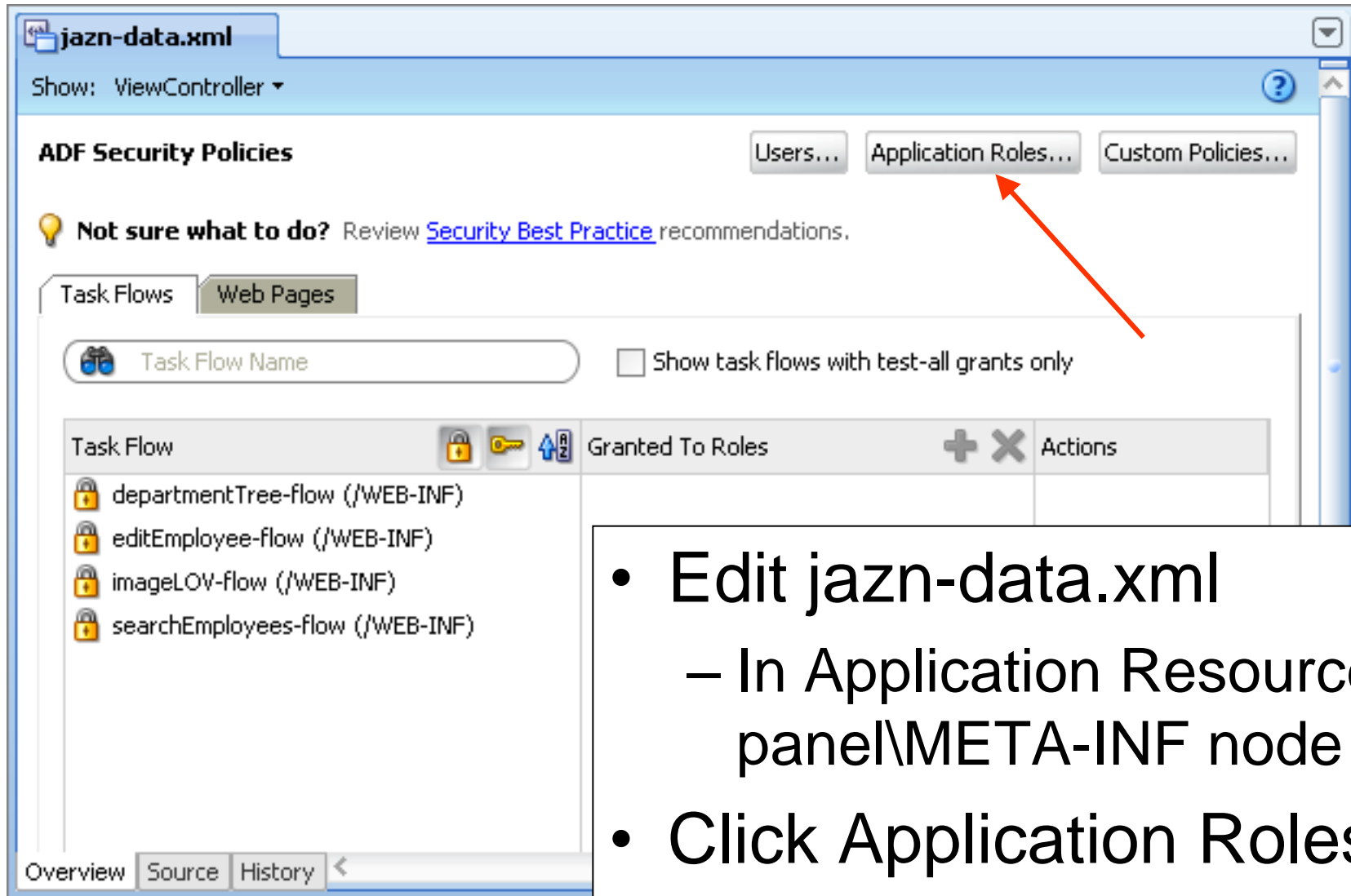


web.xml Entries - 2

```
<auth-constraint>
  <role-name>valid-users</role-name>
</auth-constraint>
</security-constraint>
<login-config>
  <auth-method>FORM</auth-method>
  <form-login-config>
    <form-login-page>/login.html</form-login-page>
    <form-error-page>/error.html</form-error-page>
  </form-login-config>
</login-config>
<security-role>
  <role-name>valid-users</role-name>
</security-role>
```



Create Application Roles



The screenshot shows the 'ADF Security Policies' configuration window for 'jazn-data.xml'. The 'Application Roles...' button is highlighted with a red arrow. Below the button, there is a search field for 'Task Flow Name' and a table listing task flows and their granted roles.

Task Flow	Granted To Roles	Actions
departmentTree-flow (/WEB-INF)		
editEmployee-flow (/WEB-INF)		
imageLOV-flow (/WEB-INF)		
searchEmployees-flow (/WEB-INF)		

- Edit jazn-data.xml
 - In Application Resources panel\META-INF node
- Click Application Roles

Edit JSP Identity and Policy Store

Application Roles

Application Roles:

- user
- manager
- admin

Buttons: Add..., Remove

Fields:

- Name: admin
- Display Name:
- Class Name: orac

Buttons: New..., Delete, Help

- Create a security policy store in Application Policy Store node
- Create application roles

Define a Credentials Store

Identity Store

Click New to create a new realm.

- Same editor, different node (Identity Store)
- Define users
- Define roles
- Map users to roles

Identity Store - Users

The screenshot shows the 'Edit JPS Identity & Policy Store' window. On the left, a tree view shows the 'Identity Store' structure, with 'Users' under 'jazn.com' selected. The main area is titled 'Users' and contains a list of users: TFOX, NKOCHHAR, and CDAVIES. The 'Add...' button is highlighted with a red arrow. Below the list are fields for Name, Credentials, Display Name, and Description. The 'Name' field contains 'CDAVIES', and the 'Credentials' field is filled with dots. At the bottom, there are 'New...', 'Delete', 'Help', 'OK', and 'Cancel' buttons.

Identity Store - Roles

The screenshot shows the 'Edit JPS Identity & Policy Store' window. The left sidebar contains a tree view with the following structure:

- Identity Store
 - jazn.com
 - Users
 - Roles**
 - Application Policy Store
 - tuhra2
 - Application Roles
 - Application Policies
 - System Policies
 - Preview XML

At the bottom of the sidebar are buttons for 'New...', 'Delete', and 'Help'.

The main area is titled 'Roles' and contains a list of roles: cs_admin, cs_manager, and cs_user. To the right of this list are 'Add...' and 'Remove' buttons. A red arrow points from the 'Add...' button to the 'Roles' list.

Below the list are three tabs: 'General', 'Member Users', and 'Member Roles'. The 'General' tab is active and contains the following fields:

- Name:
- Display Name:
- Description:

At the bottom right of the window are 'OK' and 'Cancel' buttons.

Identity Store – Users in Roles

The screenshot shows the 'Edit JPS Identity & Policy Store' dialog box. The left pane shows a tree view with 'Identity Store' expanded to 'jazzn.com' and 'Roles' selected. The main area is titled 'Roles' and contains a list of roles: 'cs_admin', 'cs_manager', and 'cs_user'. Below this list are 'Add...' and 'Remove' buttons. The 'Member Users' tab is selected, showing two lists: 'Available:' with 'NKOCHHAR' and 'CDAVIES', and 'Selected:' with 'TFOX'. Between the lists are four arrow buttons: a single right arrow, a double right arrow, a single left arrow, and a double left arrow. At the bottom are 'New...', 'Delete', 'Help', 'OK', and 'Cancel' buttons.

jazn-data.xml Contents

- Defined within a realm (namespace within the XML file)
 - By default jazn.com

Role

```
<role>
  <name>admin</name>
</role>
```

User

```
<users>
  <user>
    <name>SKING</name>
    <credentials>{903}1JHgZuUDp..
  </credentials>
  </user>
</users>
```

password
obfuscation



Users in Role

```
<role>
  <name>admin</name>
  <members>
    <member>
      <type>user</type>
      <name>TFOX</name>
    </member>
    <member>
      <type>user</type>
      <name>CDAVIES</name>
    </member>
  </members>
</role>
```

Map Application Roles to Credential Store Roles

Edit JPS Identity & Policy Store

Application Roles

Application Roles:

- user
- manager
- admin

Buttons: Add..., Remove

Tabs: General, Member Users, Member Roles

Available:

Selected:

- cs_admin
- cs_manager
- cs_user

Buttons: OK, Cancel

- Identity Store
 - jazn.com
 - Users
 - Roles
 - Application Policy Store
 - tuhra2
 - Application Roles
 - Application Policies
 - System Policies
 - Preview XML

- The LDAP repository roles will use the same names as the local store (cs_...)

Implement Authorization – Task Flow

ADF Security Policies

Users... Application Roles... Custom Policies...

Not sure what to do? Review [Security Best Practice](#) recommendations.

Task Flows Web Pages

Task Flow Name Show task flows with test-all grants only

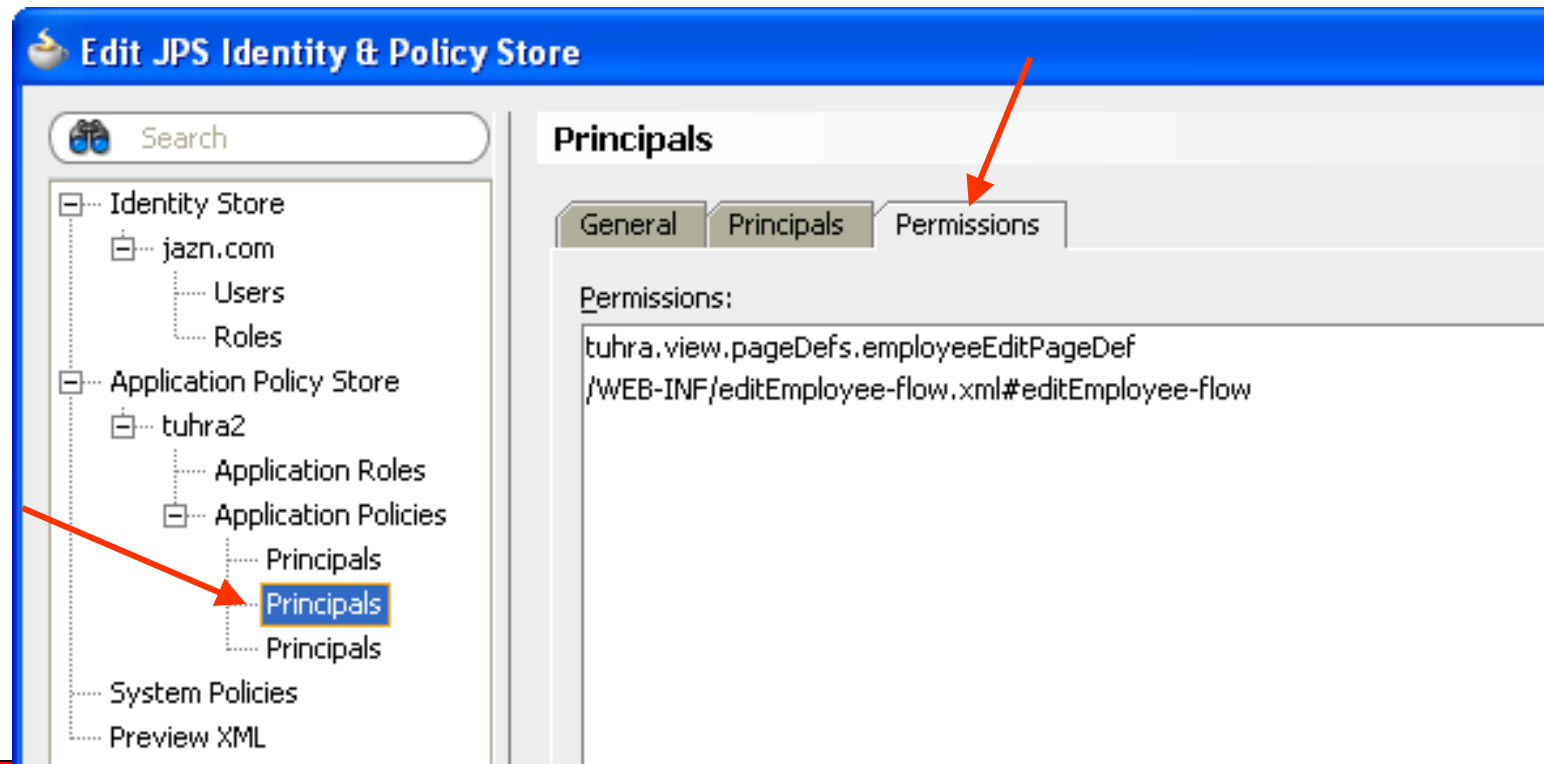
Task Flow	Granted To Roles	Actions
departmentTree-flow (/WEB-INF)	user	<input checked="" type="checkbox"/> View
editEmployee-flow (/WEB-INF)		<input type="checkbox"/> Customize
imageLOV-flow (/WEB-INF)		<input type="checkbox"/> Grant
management-flow (/WEB-INF)		<input type="checkbox"/> Personalize
searchEmployees-flow (/WEB-INF)		

Overview Source History

- Creates an ADF Security security policy

Security Policy Shown Here

- jazn-data.xml – Custom Policies button
- Each principal has a node here
- The Principals tab shows the role name (user in this case)



Implement Authorization – Web Pages

The screenshot shows the ADF Security Policies console in JDeveloper. The window title is 'jazz-data.xml'. The 'Show:' dropdown is set to 'ViewController'. The main heading is 'ADF Security Policies', with buttons for 'Users...', 'Application Roles...', and 'Custom Policies...'. A lightbulb icon indicates a tip: 'Not sure what to do? Review [Security Best Practice](#) recommendations.' Below this are tabs for 'Task Flows' and 'Web Pages', with 'Web Pages' selected. A message states: 'Web pages must have an associated page definition file to be directly securable.' There is a search box for 'Page Name' and a checkbox for 'Show web pages with test-all grants only'. A table lists page definitions and their security grants.

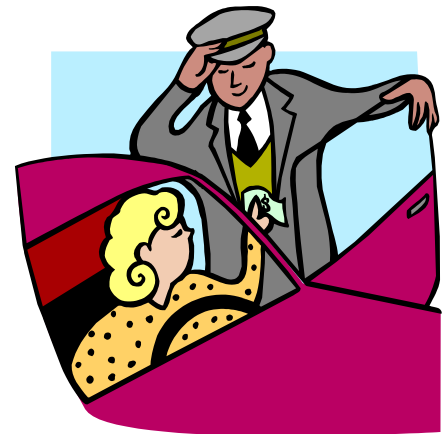
Page Definition	Granted To Roles	Actions
employeeEdit (tuhra.view.pageDefs)	user	<input checked="" type="checkbox"/> View <input type="checkbox"/> Customize <input type="checkbox"/> Edit <input type="checkbox"/> Grant <input type="checkbox"/> Personalize
employeeSearch (tuhra.view.pageDefs)		
management (tuhra.view.pageDefs)		
templateTest (tuhra.view.pageDefs)		
tuhraTemplate (templates)		
tuhraTrainTemplate (templates)		

At the bottom, there are tabs for 'Overview', 'Source', and 'History'.

- Creates or edits a security policy

Development Tips

- Setting in JDev declares if security settings will be refreshed
 - Usually no need to stop and restart the application when testing security
 - Stopping/starting undeploys and redeploys the application
 - There is also no need to stop the Default Server
- You might need to exit the browser session before running a new version of the application



Agenda

- Overview
- Application security tasks
- UI techniques



Example – Hiding a Link

- Hide the item if the user is not authorized to view the management-flow
 - More generic – based on permission rather than a role name (permissions may be time or location sensitive)

```
<af:commandLink rendered=
"#{securityContext.taskflowViewable[
  '/WEB-INF/management-flow.xml#management-flow']}"
```

- Alternative: Hide the item if the user is not in the manager role

```
<af:commandLink rendered=
#{securityContext.userInRole['manager']}
```



Example – Disabling a Menu Item

- Use general Security Context property on the Disabled property
 - securityContext.authenticated

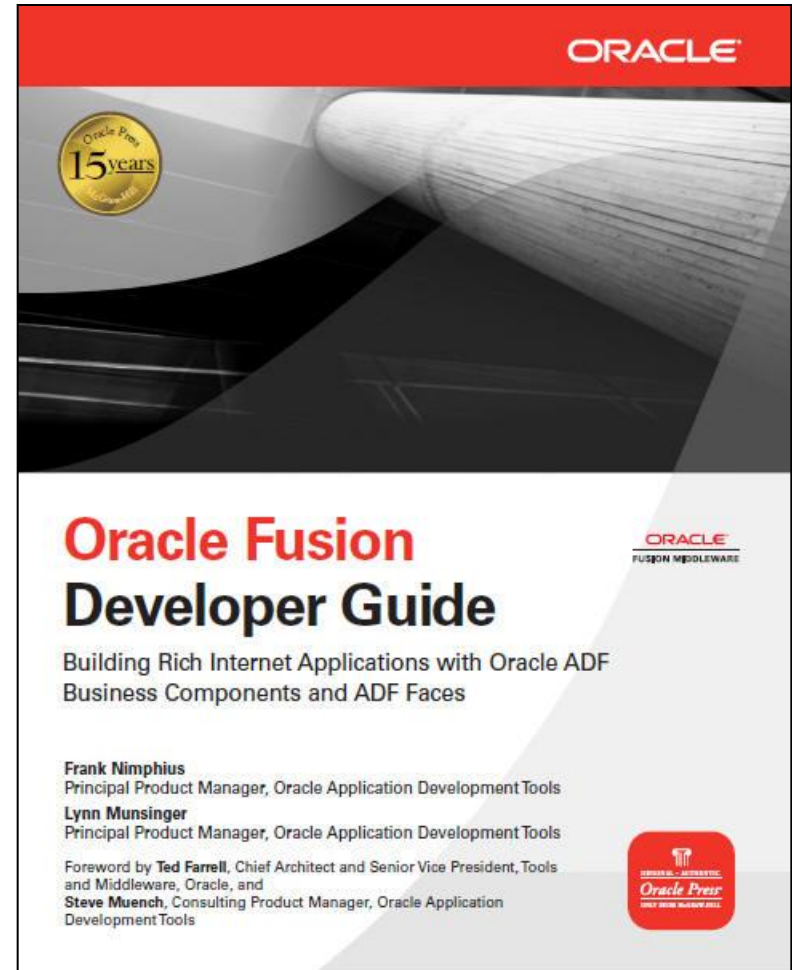
```
<af:commandMenuItem text="Edit My Info" id="pt_cmi5"  
disabled="#{!securityContext.authenticated}">
```

- If the user is not authenticated (successfully logged in), this menu item will be grayed-out (disabled)



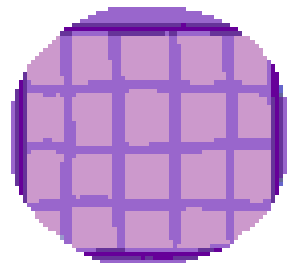
Other Resources

- *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework 11g Release 1 (11.1.1)*
 - PDF at OTN, online in JDev
 - Chapter 29
- *Oracle Fusion Middleware Security Guide 11g Release 1*
 - PDF at OTN
- *Oracle Fusion Developer Guide*
 - Nimphius and Munsinger, McGraw-Hill Professional, Oracle Press (2010)
 - Chapter 21

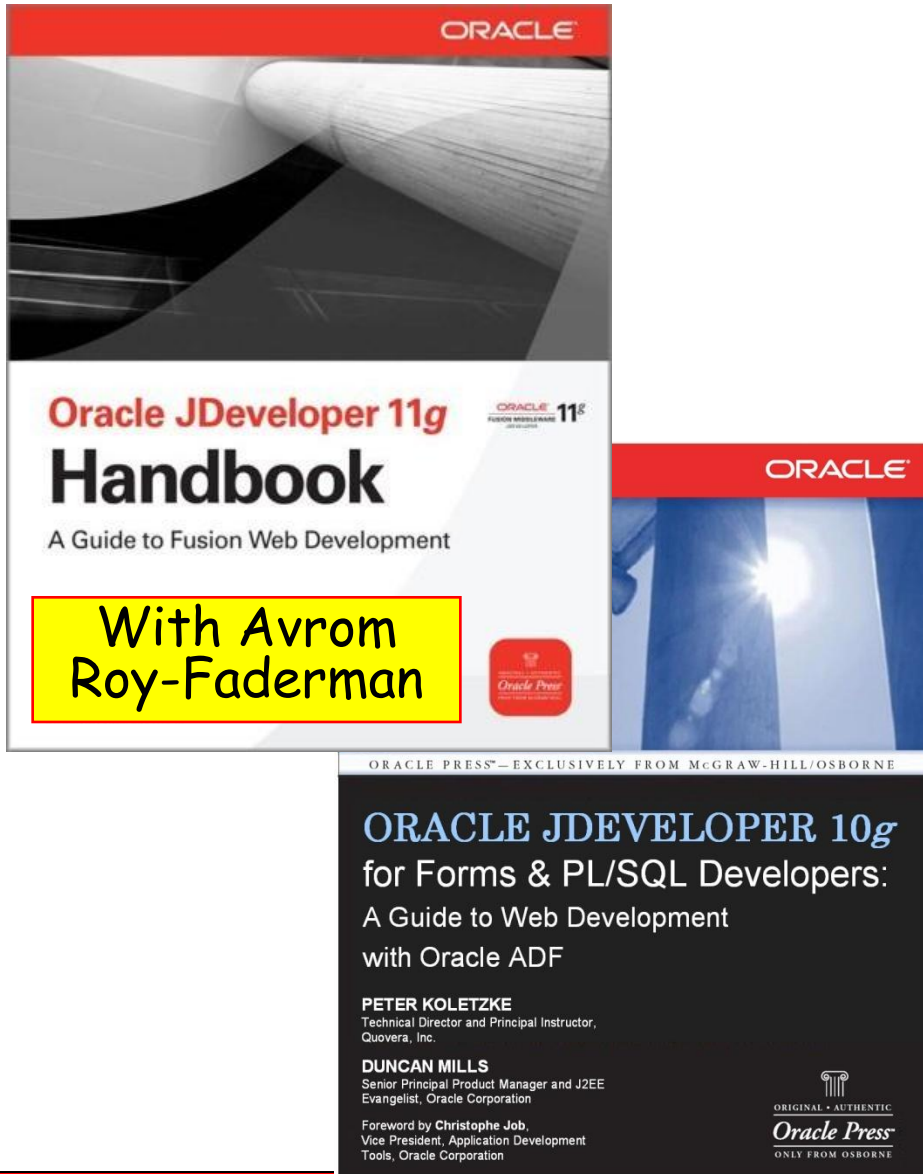


Summary

- You need to design application security
- OPSS offers easy access to standard JAAS security features
- ADF Security provides declarative definition of security policies for task flows and pages
- Binding expressions on the page can hide or disable items
- Design and test for all security breach scenarios



The Books



The Coauthors

- Duncan Mills
 - Widely published on OTN, ODTUG, JDJ etc.
 - groundside.com/blog/DuncanMills.php
 - www.oracle.com
- Peter Koletzke
 - Six other Oracle Press books about Oracle tools
 - www.quovera.com
- Book examples
 - www.tuhra.com