HANDS-ON PRACTICE 2
PRIVILEGES FOR ADF BC OBJECTS

Peter Koletzke, Quovera
Duncan Mills, Oracle Corp.

Just as you can declare hooks to security roles using settings in the user interface components, you can alternatively or also set properties on the Application Development Framework Business Components (ADF BC) entity object. Once you set the security properties in this way, all user interface objects built from that entity object (through a view object) will immediately be affected. For example, if you wanted to be sure that the employee's salary is enabled for update if the user is in the admin or manager roles. You can set attributes or the entire entity object to read-only, update while new, or update always for specific roles.

The following steps demonstrate how to set up security on the ADF BC entity object. The practice is taken in part from the *Oracle JDeveloper 10g for Forms and PL/SQL Developers* (Oracle Press, McGraw-Hill/Osborne).

1. Enable ADF BC security by opening the configuration (select Configurations from the right-click menu on the application module and clicking Edit), selecting the Properties tab, and entering the jbo.security.enforce value as "Auth" (be careful to match case and spelling) as shown in Figure 1. After changing the property, click OK and OK to dismiss the editor.

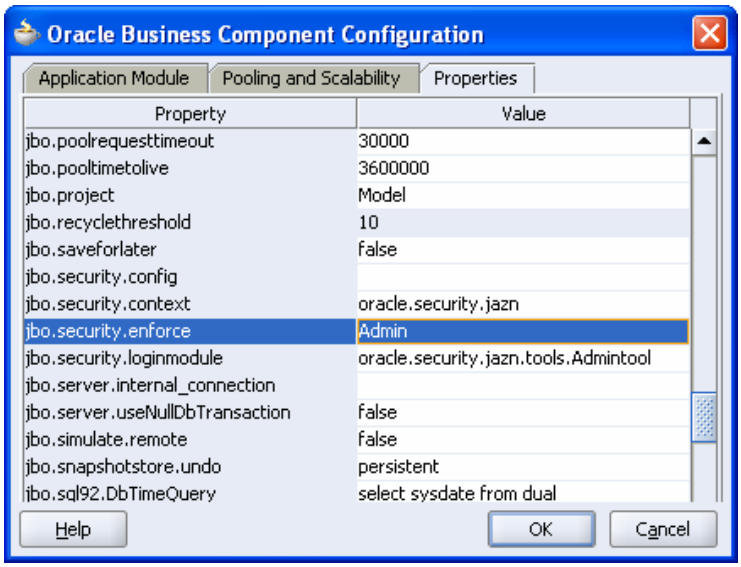

Figure 1. Configuration setting to enable ADF BC security

2. The next several steps require manually editing the JAZN files in your local OC4J instance. This is only necessary for testing purposes. When your application is deployed to a production environment, all users will already be set up on the JAZN files in the server's OC4J instance. Assuming your workspace is called "Employees" (as in the first practice), open the Employees-jazn-data.xml file in the workspace directory. Copy the user definitions that you added to the workspace (following the example in the hands-on practice, these are AHUNOLD, CDAVIES, SKING, and SKUMAR). Be sure to include all starting and ending "<user>" tags.

3. Open the file JDEV_HOME/ jdev/ system /oracle.j2ee.10.1.3.n.n/ embedded-oc4j/ config/ system-jazn-data.xml (where JDEV_HOME is the directory into which you installed JDeveloper). Find the same section for users under the <jazn-realm> element and paste the user definitions in the clipboard there. Double check that all tags are matched correctly and are embedded within the correct parent. Repeat the copy and paste operation for roles from the workspace JAZN file to the JDeveloper system JAZN file.

4. Repeat the copy and paste operations for users and roles into the file. Triple check that each file has the new roles and users and that the tags are matched and are within the correct parent tag.
5. Close JDeveloper if it is open and reopen it. Double click the entity object (for example, Employees) and navigate to the Authorization node. Select Salary and click New. The Authorization dialog shown in Figure 2 will appear. Set the admin role to "Update Permission" and click OK. Click New and repeat this setting for the manager role. Click New and set the user role to Read-Only.
6. Select the entity object level node in the Authorization page and set constraints of "Update Permission" for all roles. The attribute-level authorizations for Salary will override the entity object-level authorizations but all other attributes will have update allowed. The Authorizations page permissions should appear as shown in Figure 2.

   Note: Remove any JSP component EL expressions for *Enabled* or *Rendered* before testing the ADF BC permissions.
7. When you run any page that is connected to a view object, which is based on this entity object, the UI will be drawn accordingly. For example, manager and admin roles will always see the Salary item but user role users will not.

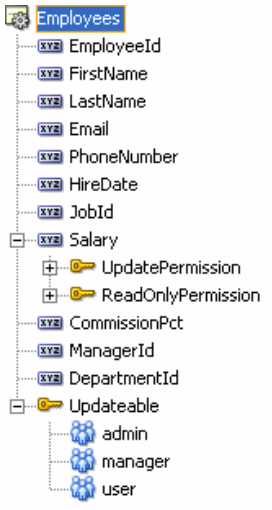

Figure 2. Entity object authorizations for Employees