Protecting Access to Application Services in CE 7.1

Applies to:

SAP NetWeaver Composition Environment 7.1 SR3 Evalution SDN Version. For more information, visit the Composition homepage.

Summary

This article explains how to implement permission checks for Application Services in CE 7.1.This article is based on the tutorial “Service Composition with SAP Composite Application framework Capabilities in SAP NetWeaver 7.1”.It implements the permission checks for the Project Manager Service operations.

Author: Sampath Gunda

Company: HCL Technologies, Chennai, India.

Created on: 30 June 2008

Author Bio

Sampath Gunda is working as a Java Developer in HCL Technologies, Chennai, India.

Table of Contents

Protecting Access to Application Service Operations

You can control which users and groups are allowed to execute application services' operations by implementing UME permission checks. This mechanism uses standard Java permissions and allows you to implement custom logic for verifying user authorizations, as you do for standard Java Enterprise Edition (EE) applications.

Before beginning with this article, you should be familiar with UME roles, actions, and permissions.

Permissions, Actions, and UME Roles

Definition

Authorizations are enforced in the user management engine (UME) using permissions, actions, and roles.

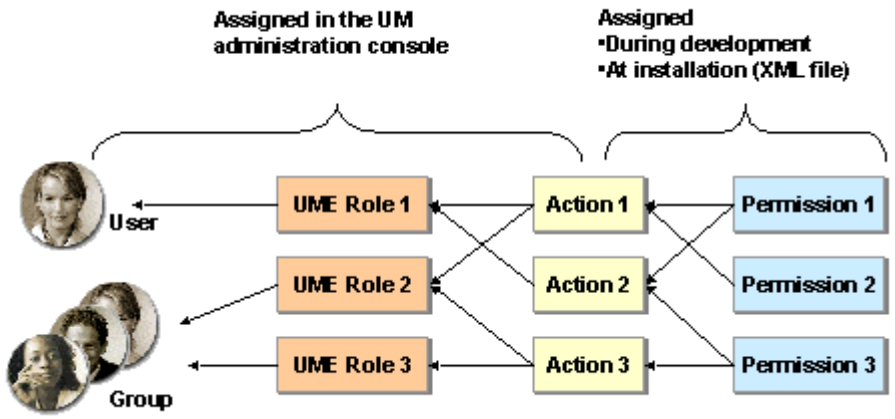
Internally in their Java code, applications define UME **permissions** and use them for access control. UME permissions are an implementation of Java permissions.

An **action** is a collection of permissions. Every application defines its own set of actions and specifies the permissions assigned to the actions either in an XML file or (more seldom) dynamically in the code. The actions appear in the user management administration console, where you can group them together into roles.

UME Roles group together actions from one or more applications. You assign roles to users in the user management administration console. By assigning roles to users, you define the users' authorizations.

Structure

The following figure illustrates the relationship between permissions, actions, and roles.



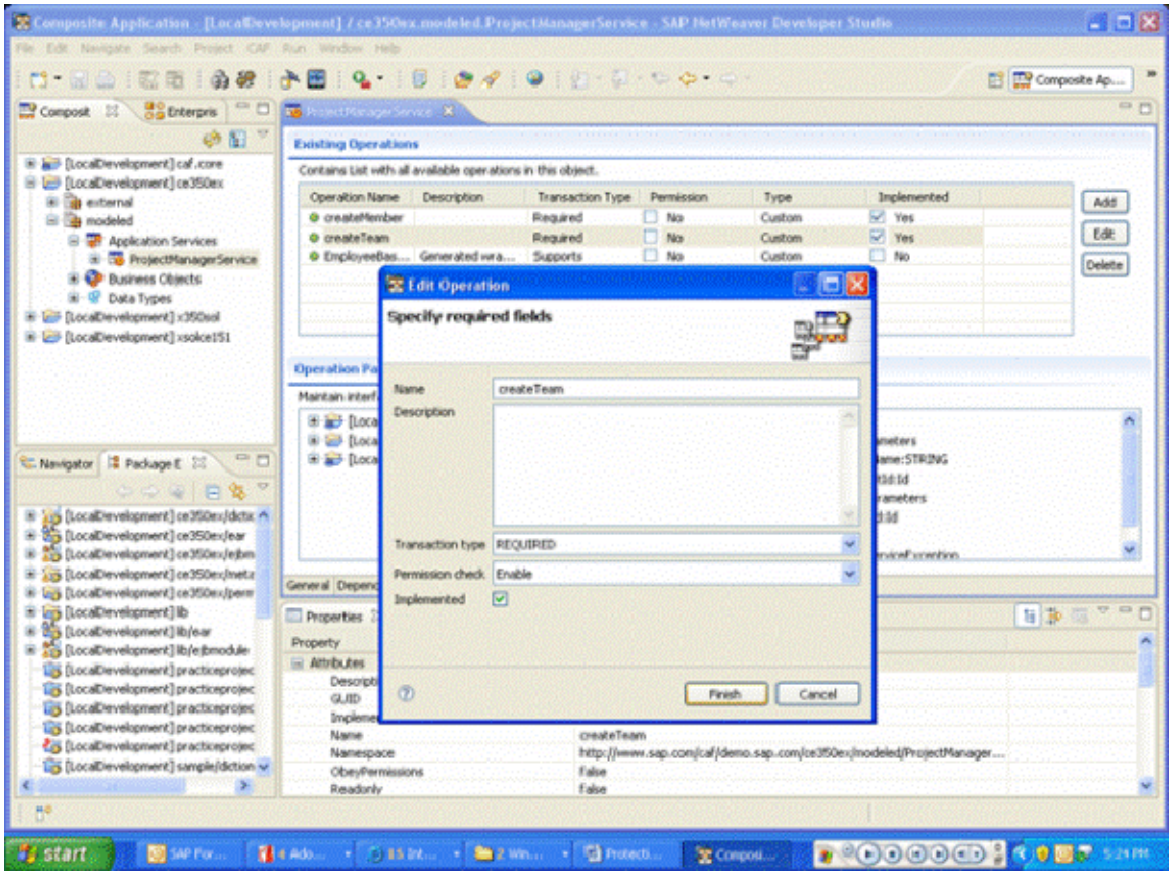
The advantage of having both actions and permissions is:

- Application developers can define finely grained permissions, but can hide the complexity by defining only a few actions.
- As the actions are normally defined in an XML file, they can be changed according to your requirements when you install the service.
- Administrators can assign actions to roles in the administration console. Permissions are not visible in the administration console.

Implementing Permission checks for Application services

Let us implement the Permission checks for ProjectManager application service of the example CAF application **ce350ex** which is created in the tutorial "Service Composition with SAP Composite Application framework Capabilities in SAP NetWeaver 7.1". You can download this tutorial using this URL.

In the composite application perspective of NetWeaver Developer Studio, select the CAF Application (ce350ex) and open the Project Manager Application Service. Switch to the Operations tab. Select the createTeam operation and click edit button.

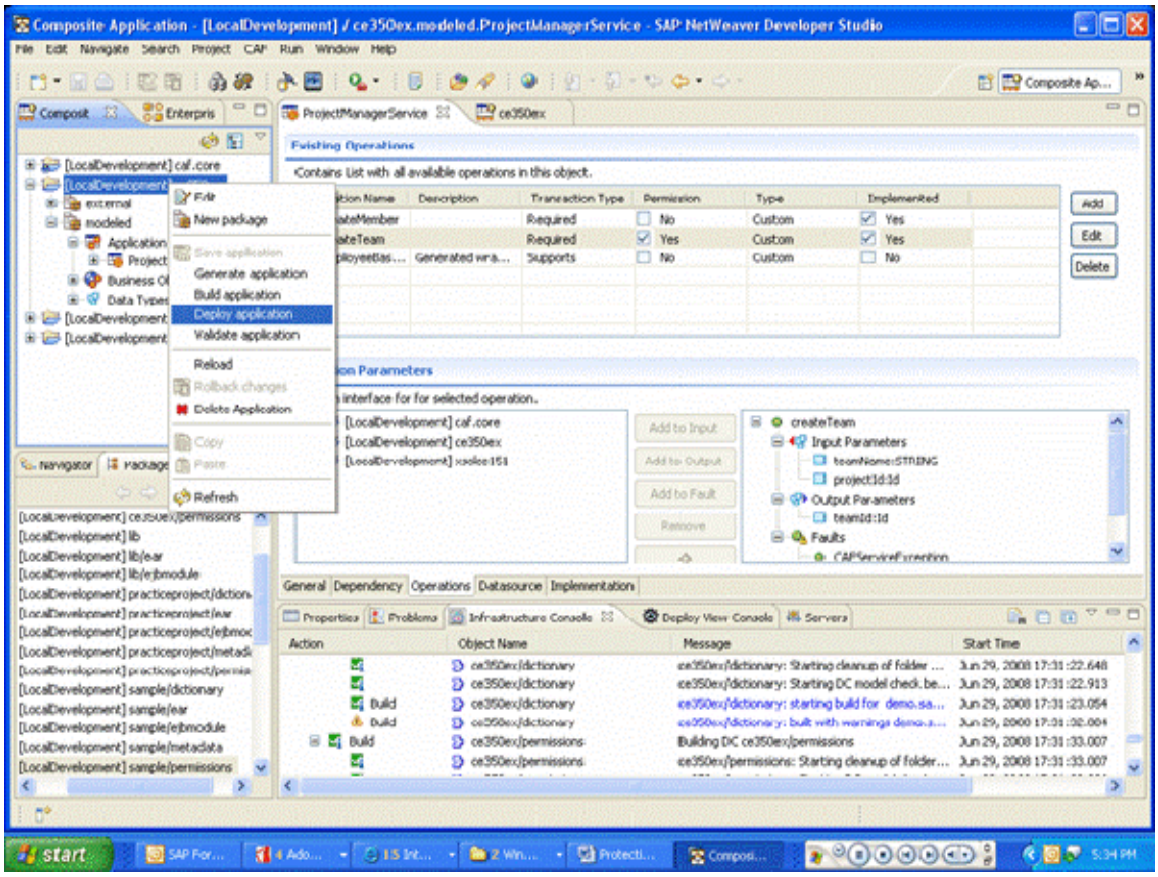


Select the "Enable" option from Permission check drop down box. Click Finish button.

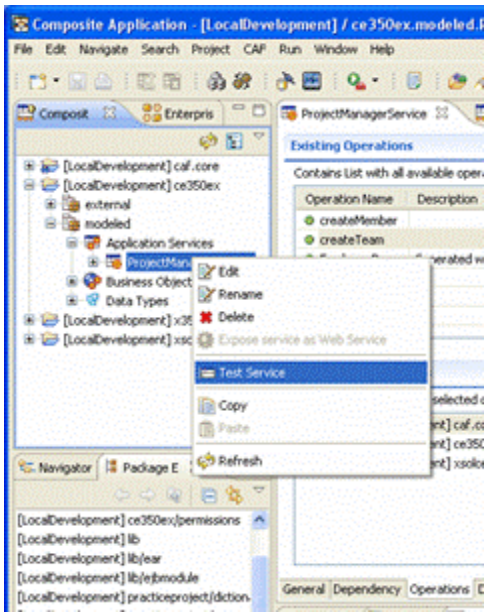
A new permission with permission class com.sap.caf.rt.security.srv.ServicePermission is generated. It extends the java.security.BasicPermission class and is automatically used as the permission class for the permissions you define in your CAF application.

Similarly, enable Permission Check option for createMember operation also.

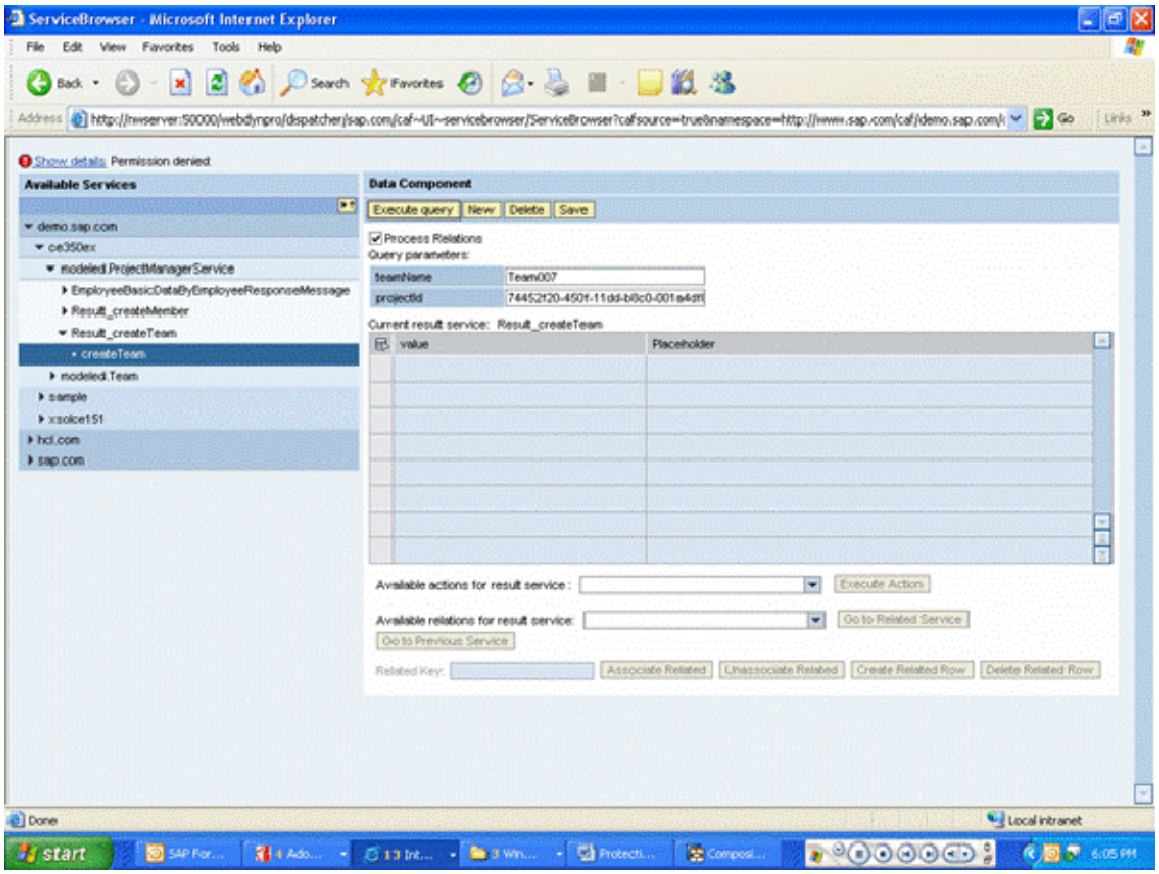
Generate, build and deploy the application.



Now try to create a team using createTeam operation in the Service Browser window. To open the Service Browser, right click on the ProjectManager application service and select Test Service (You may need to login with Administrator credentials).



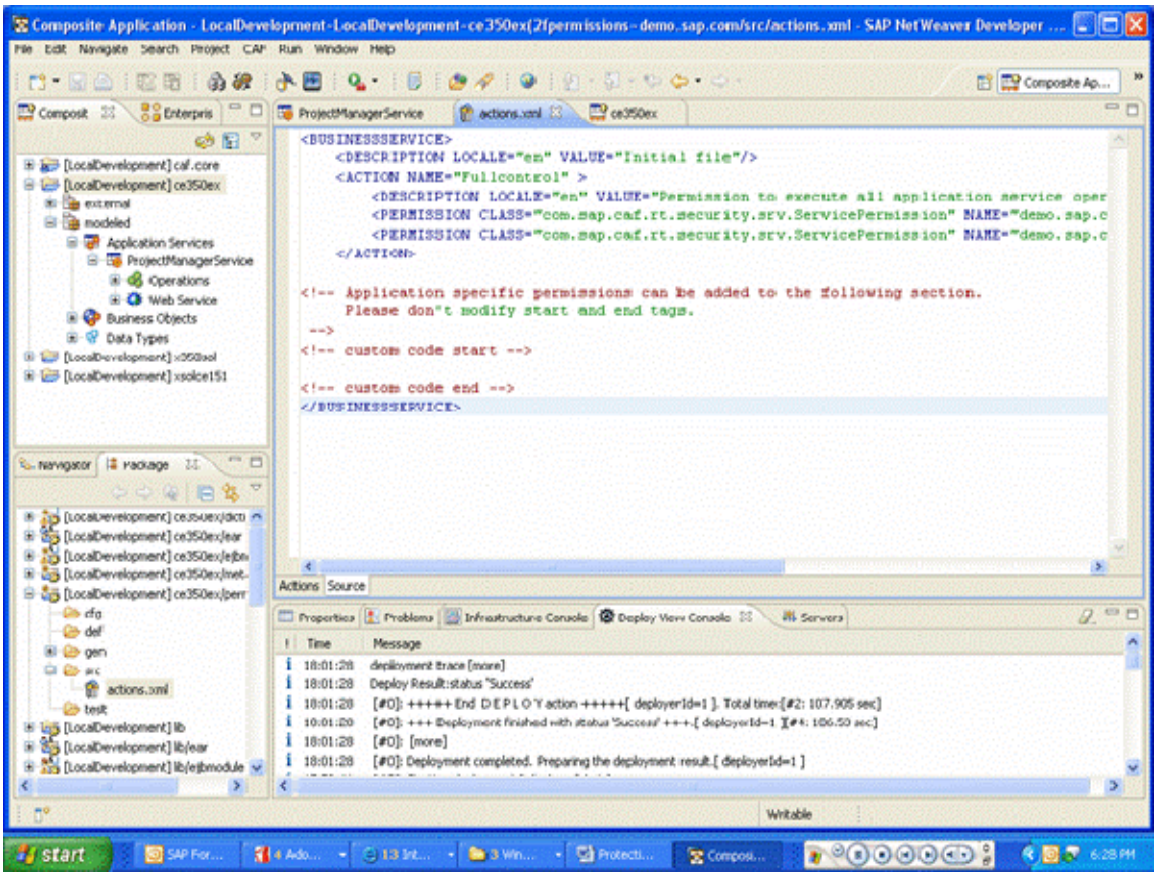
When you try to access the createTeam method, you will receive an error message such as"Permission denied".



This is caused by the new Permission settings you have made for createTeam operation of ProjectManager application service. Permission checks are done at runtime.

The following section explains the procedure to setup the permissions for different users.

In the Permission DC of your CAF Application, you can find a file named Actions.xml, located in folder /src. Open the Actions.xml file.



By default, this file contains one action entry having one permission for application service operation with permission checks enabled. Each permission is named by the fully qualified service name and operation name with the format : **<provider>/<xapp>/<service name>/<operation name>.**

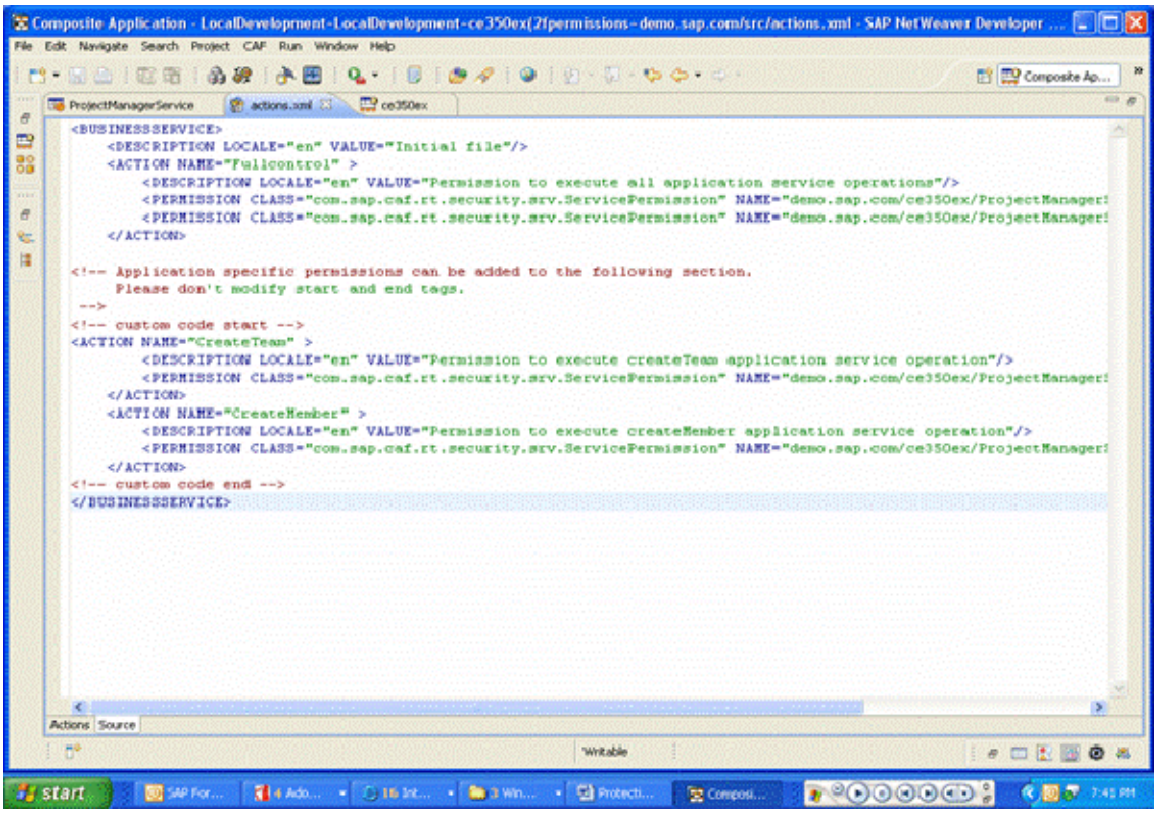
Ex: demo.sap.com/ce350ex/ProjectManagerService/createTeam

In our case, the “Full Control” action contains two permissions one for createMember operation and another one for createTeam operation.

If you deploy the Permissions DC’s EAR file, you can find the mentioned “full control” action in the User Management tool. When assigning this action to a UME role, each user assigned to that role will be allowed to access all the operations of your application service. This is typically undesirable.

Therefore you should add an action per operation to the “custom code” of action.xml file. Use the “Full Control” as a template and name your action exactly like the operation it is referring to.

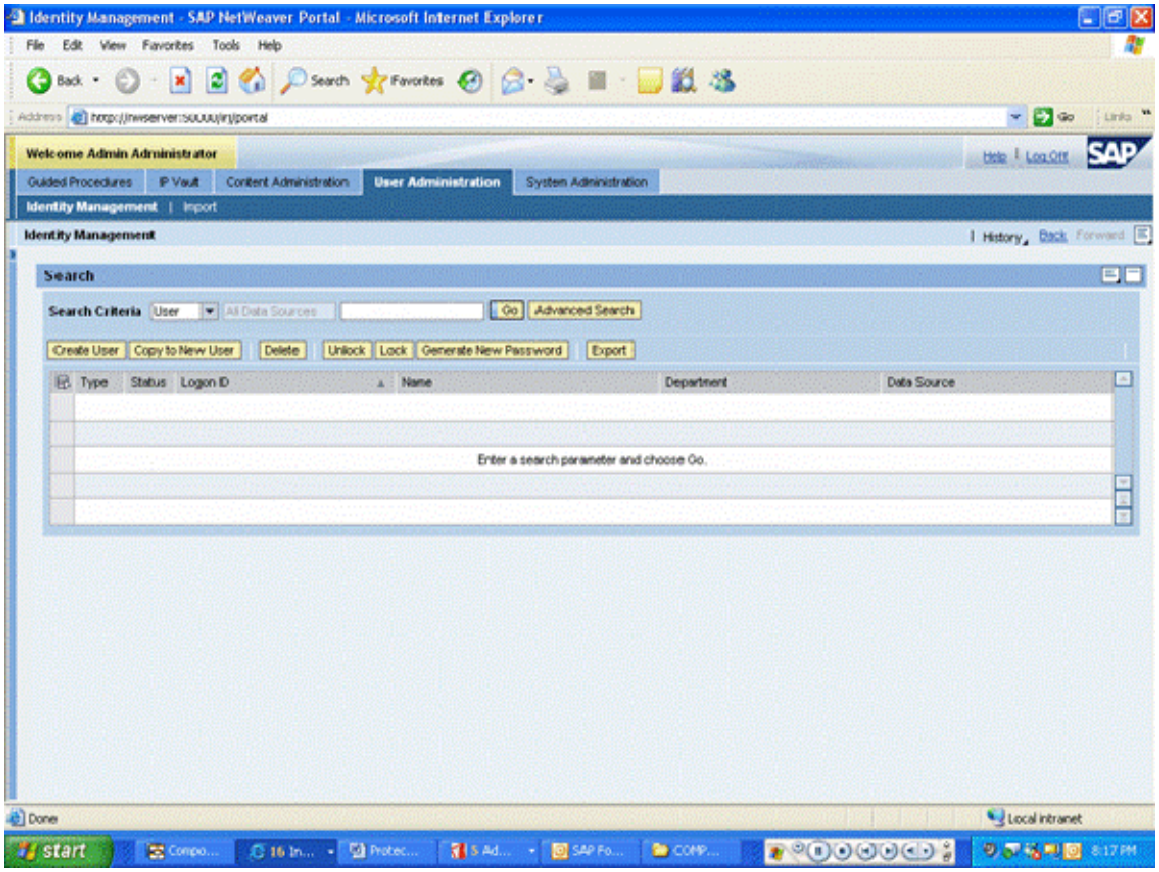
Add two actions CreateTeam and CreateMember to the “custom code” section of the actions.xml file.



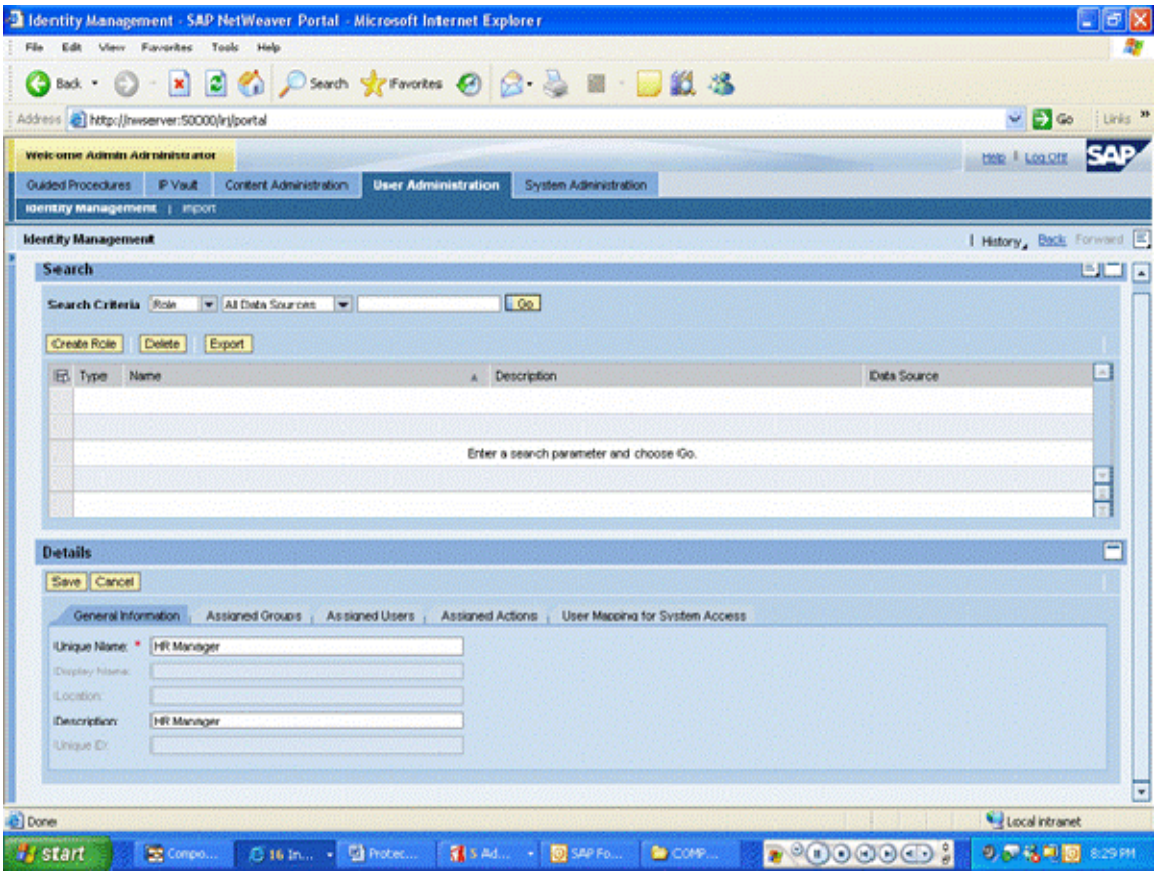
Save the file. Generate, build and deploy the CAF Application.

Once the actions have been deployed on the AS Java, you have to create the UME roles that contain the actions you have just created and assign them to the required users.

Open the NetWeaver Portal with the URL http://<nwserver>:50000/irj/portal. Log in to the NetWeaver Portal with Administrator account and select the **User Administration** tab.



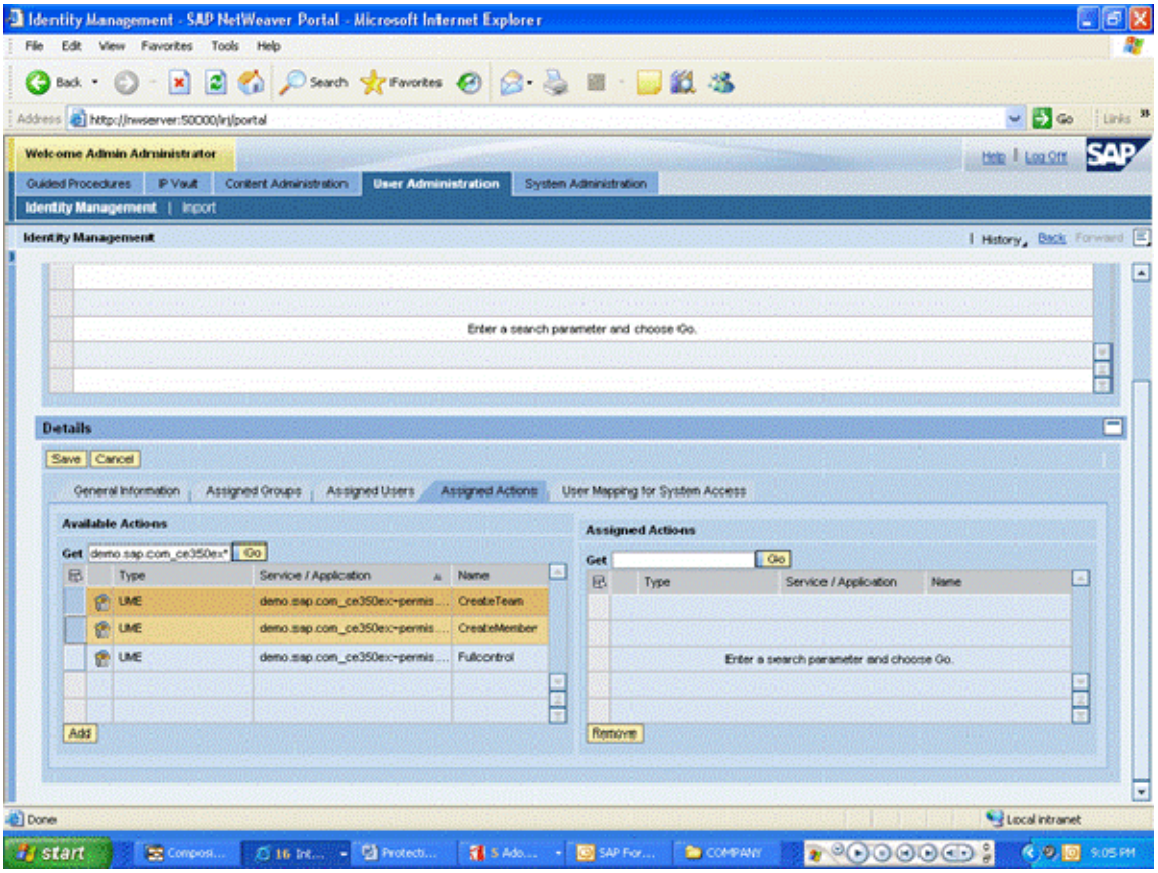
In the Identity Management console, select the “Role” option from the “Search Criteria” drop down box. Choose Create Role. The details will appear in the lower section of the Web browser.



In the **General Information** tab enter unique name as “HR Manager” and fill up the optional Description column.

Select the Assigned Actions tab page.

In the available Actions section, enter **demo.sap.com_ce350ex*** in the Get field and click Go. It will list the actions associated with **ce350ex** application.

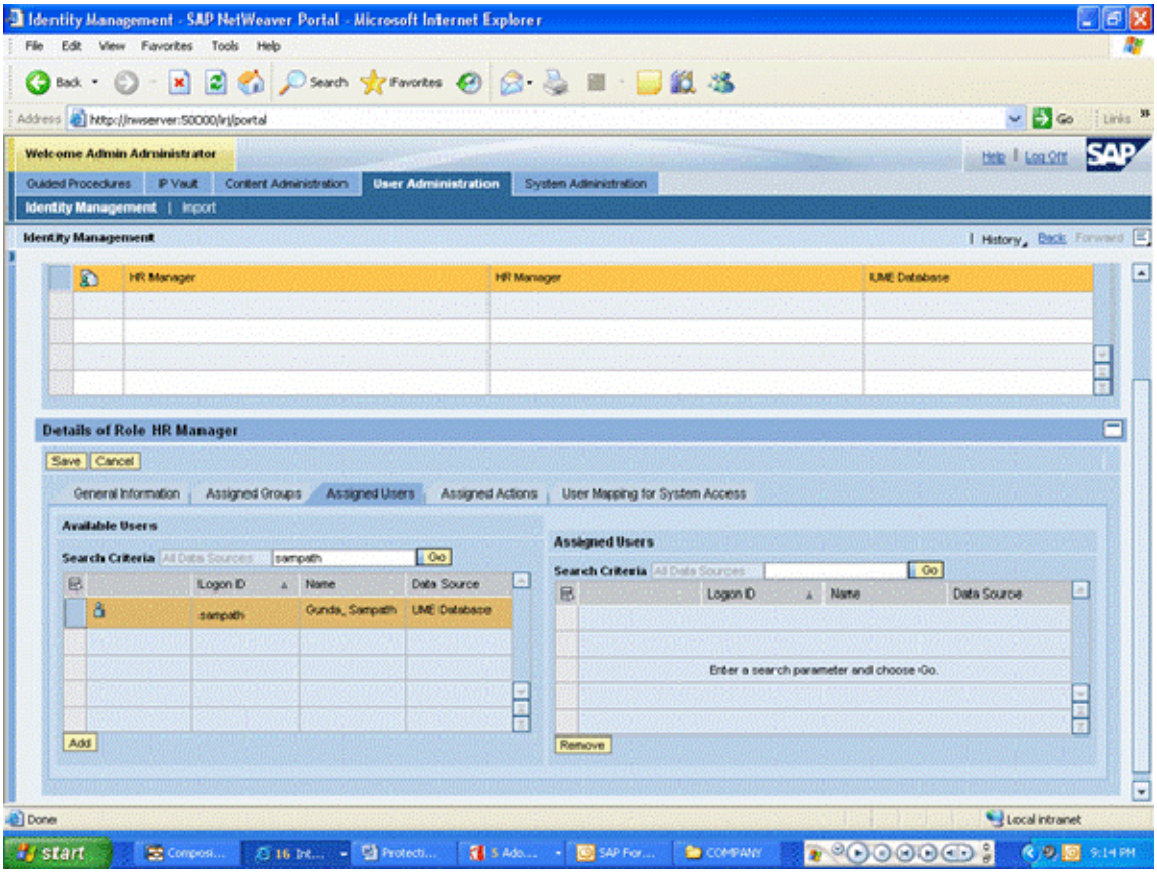


Select the actions **CreateTeam** and **CreateMamber** and choose Add. The selected actions will be added to the Assigned actions table. Save the data.

Now assign the required users to the role. If the corresponding users are all ready available on the Portal, you can assign the actions to them. Otherwise, you have to create the user accounts using the **Create User** option in the **Identity Management** console. For our case, the user is all ready available on the portal.

Choose the Assigned Users Tab and click the Modify button.

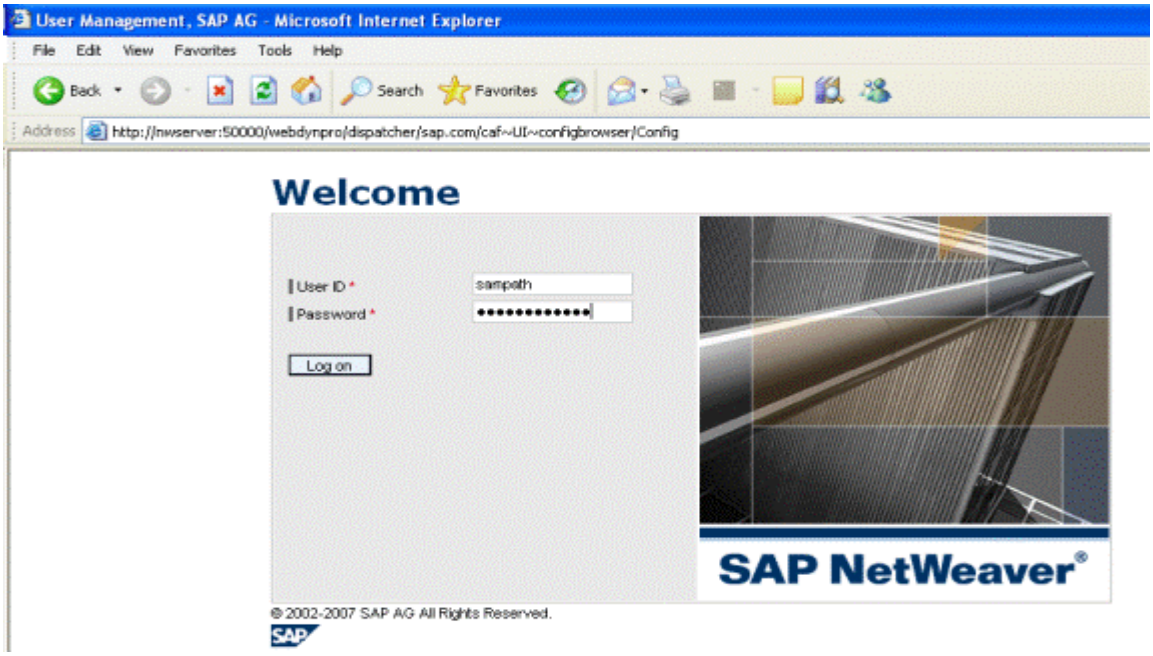
In the Available Users section, enter the user ID of the corresponding user to search for in the *Get* field and choose *Go*. It will display the corresponding user in the Available Users table.



Select the user and click *Add.* The user is added to the Assigned Users section. Save the data.

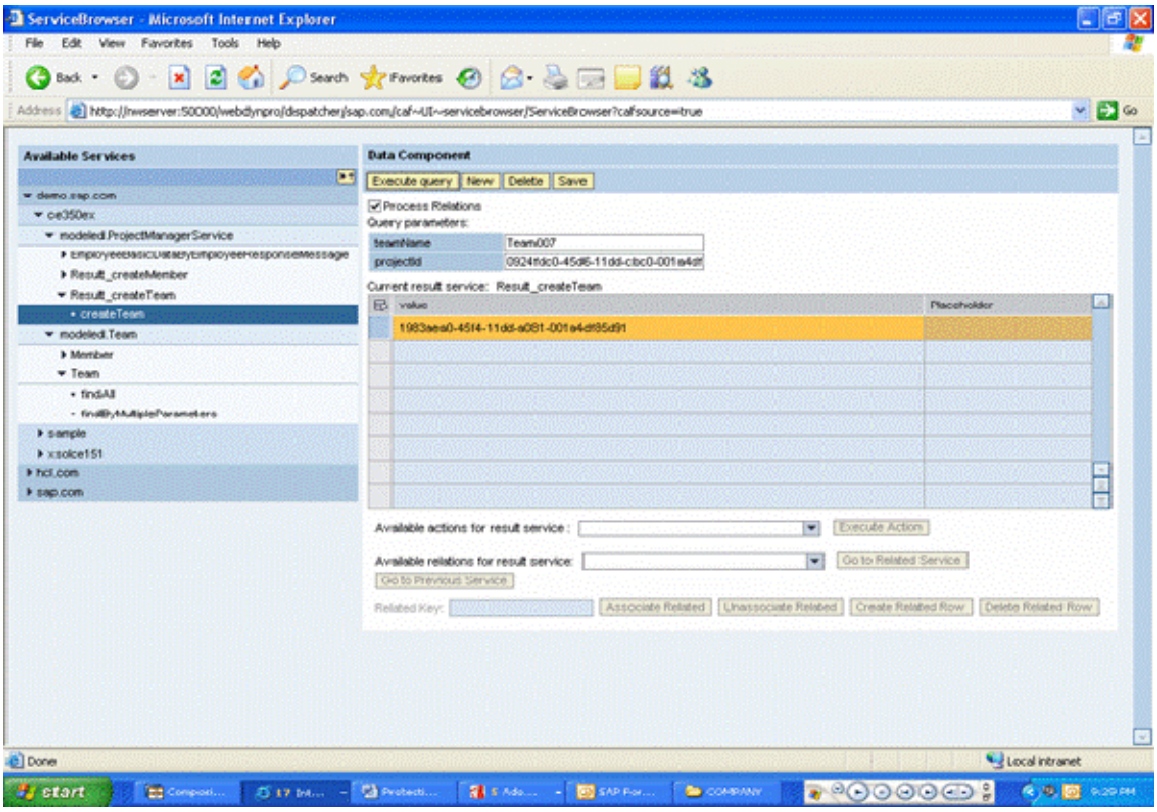
The HR Manager role is created with the actions you have created and assigned to the userId “Sampath”. Now the user “Sampath” can execute the application service operations createTeam and createMember.

Open the Service Browser with URL: **http://<host name> :< port number>/caf** and login with the sampath and test the operations.



It allows the user (Sampath) to create a Team instance with createTeam operation.

Similarly you can check the createMember operation also.



Alternatively, you may implement application service operations permission checks by coding in the operation itself.

Related Content

Service Composition with SAP Composite Application Framework Capabilities in SAP NetWeaver CE 7.1

Modeling Application Services (SAP Library)

SAP NetWeaver Composition Environment 7.1 Tutorial Center

Architecture-Guidelines Part 3: Business Logic, Abstraction Layer and Connectivity

For more information, visit the Composition homepage.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.