Radio Buttons and Check Boxes as Table Selectors and UME API Usage in Web Dynpro

Applies to:

Web Dynpro Java Development, SAP NetWeaver 2004s. For more information, visit the User Interface Technology homepage.

Summary

In this tutorial you can learn how to set the radio buttons, check boxes as table selectors for single and multiple selections, and also the UME operations like portal users search, role assignments using UME API in Web Dynpro for Java. Download source code file from here.

Author: Naga Raju Meesala

Company: Robert Bosch Engineering and Business Solutions Ltd.

Created on: 04 September 2008

Author Bio

Naga Raju Meesala works as a SAP EP consultant in BOSCH Group. His main responsibilities are the Development of Web Dynpro for Java components and integration with the SAP NetWeaver Portal and Portal customization

Table of Contents

Introduction

This Application is going to explain an application for assign and unassigned the roles for a portal user.

The topics going to be covered here are:

1. Portal Users Search using UME API.
2. Radio Buttons as the Table row selector.
3. Check boxes as the Table row selector.
4. Assign and unassigned the roles to portal users using UME API.
5. Search in table.

Web Dynpro Application Structure

A Simple Web Dynpro project structure contains a component and a view.

Project Name: RoleAssignWD

Application Name: RoleAssignUnAssign

Component Name: RoleApp

Window Name: RoleComp

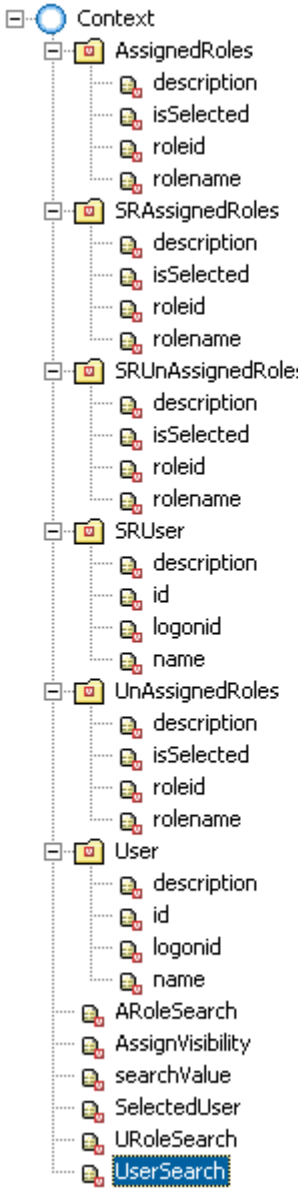
View : RoleView

Jar files required: com.sap.security.api.jar

Component Controller Implementation

Define the context as shown below

The Context nodes below User, AssignedRoles and UnAssignedRoles to contain the data of users and roles retrieved from the portal. Nodes like SRUser, SRAssignedRoles and SRUnAssignedRoles are the duplicate nodes to above specified nodes respectively. These are useful when implementing the search functionality. Coming to attributes SearchValue, UserSearch, ARoleSearch and URoleSearch are to bind to the input fields to take the input. AssignVisibility is used to hide or show the operations pane. SelectedUser Attribute is for to identify which radio button of the table selected.



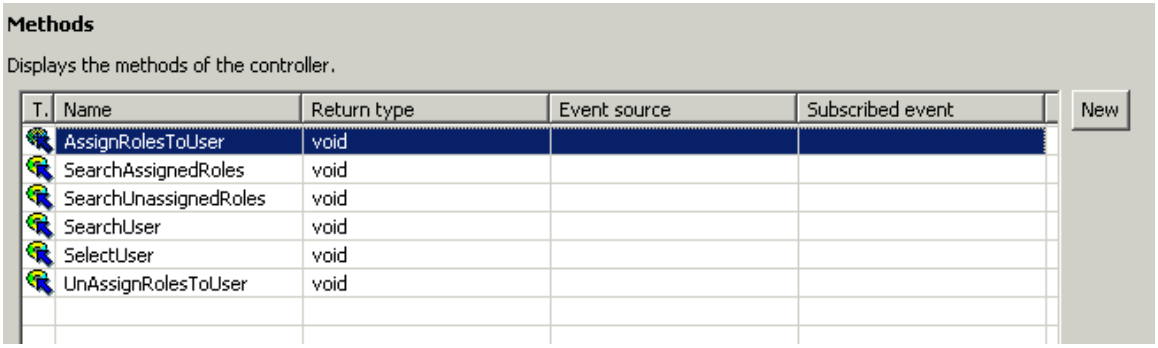
All nodes cardinality is 0..n and selection is 1..1

isSelected attribute in all nodes is of type boolean.

The attribute AssignVisibility is type of com.sap.ide.webdynpro.uielementdefinitions.Visibility.

Remaining all are of type String.

Defines the public methods in controller as shown below



No method contains parameters.

Here is the Implementation:

```
//@@begin imports
import java.util.HashSet;
import java.util.Iterator;

import com.rbei.bmh.ume.wdp.IPrivateRoleApp;
import com.rbei.bmh.ume.wdp.IPublicRoleApp;
import com.sap.security.api.IRole;
import com.sap.security.api.IRoleFactory;
import com.sap.security.api.IRoleSearchFilter;
import com.sap.security.api.ISearchAttribute;
import com.sap.security.api.ISearchResult;
import com.sap.security.api.IUser;
import com.sap.security.api.IUserFactory;
import com.sap.security.api.IUserSearchFilter;
import com.sap.security.api.UMException;
import com.sap.security.api.UMFactory;
import com.sap.tc.Web Dynpro.progmodel.api.IWDAttributeInfo;
import com.sap.tc.Web Dynpro.progmodel.api.IWDMessageManager;
import com.sap.tc.Web Dynpro.progmodel.api.IWDNode;
import com.sap.tc.Web Dynpro.progmodel.api.IWDNodeElement;
import com.sap.tc.Web Dynpro.progmodel.api.WDCopyService;
import com.sap.tc.Web Dynpro.progmodel.api.WDVisibility;
//@@end

public void wdDoInit()
  {
    //@@begin wdDoInit()
   messageManager=wdComponentAPI.getMessageManager();
    //@@end
  }
public void AssignRolesToUser( )
  {
    //@@begin AssignRolesToUser()
 try {
  String selUser =wdContext.currentContextElement().getSelectedUser();
  IRoleFactory fact = UMFactory.getRoleFactory();
  String id= UMFactory.getUserFactory().
             getUserByLogonID(selUser).getUniqueID();
  int size = wdContext.nodeUnAssignedRoles().size();
  for (int i = 0; i < size; i++) {
   if (wdContext.nodeUnAssignedRoles().
       getUnAssignedRolesElementAt(i).getIsSelected()) {
   fact.addUserToRole(id,wdContext.nodeUnAssignedRoles().
                         getUnAssignedRolesElementAt(i).getRoleid());
   }
  }
  messageManager.reportSuccess("Assigned Roles Successfully");
  getAssignedRoles(selUser);
  getUnAssignedRoles();
  WDCopyService.copyElements(wdContext.nodeAssignedRoles(),
                         wdContext.nodeSRAssignedRoles());
  WDCopyService.copyElements(wdContext.nodeUnAssignedRoles(),
                        wdContext.nodeSRUnAssignedRoles());
  wdThis.SearchAssignedRoles();
```

```
  wdThis.SearchUnassignedRoles();
 } catch (UMException e) {
  messageManager.reportException(e.getMessage(), false);
 }
//@@end
}


  public void SearchUser( )
  {
    //@@begin SearchUser()
   wdContext.currentContextElement().setAssignVisibility(WDVisibility.NONE);
   String searchCriteria = wdContext.currentContextElement().getSearchValue();
   if (searchCriteria == null || searchCriteria.trim().equals(""))
      searchCriteria = "*";
   wdContext.currentContextElement().setSelectedUser(null);
   wdContext.nodeUser().invalidate();
   try {
         IUserFactory userFactory = UMFactory.getUserFactory();
         IUserSearchFilter searchfilter = userFactory.getUserSearchFilter();

   searchfilter.setUniqueName(searchCriteria,ISearchAttribute.LIKE_OPERATOR,false);
         ISearchResult searchResult = userFactory.searchUsers(searchfilter);
         while (searchResult.hasNext()) {
               IUser user = userFactory.getUser((String) searchResult.next());
               IPublicRoleApp.IUserElement
userElement=wdContext.createUserElement();
               userElement.setId(user.getUniqueName());
               userElement.setLogonid(user.getUniqueID());
               userElement.setName(user.getDisplayName());
               userElement.setDescription(user.getEmail());
               wdContext.nodeUser().addElement(userElement);
         }
   } catch (UMException e) {
         messageManager.reportException(e.getMessage(), true);
   }
    //@@end
  }

  //@@begin javadoc:SearchAssignedRoles()
  /** Declared method. */
  //@@end
  public void SearchAssignedRoles( )
  {
    //@@begin SearchAssignedRoles()
   filterTable(wdContext.nodeAssignedRoles(),wdContext.nodeSRAssignedRoles(),wdContex
t.currentContextElement().getARoleSearch());
    //@@end
  }

  //@@begin javadoc:SearchUnassignedRoles()
  /** Declared method. */
  //@@end
  public void SearchUnassignedRoles( )
  {
    //@@begin SearchUnassignedRoles()
```

```
   filterTable(wdContext.nodeUnAssignedRoles(),wdContext.nodeSRUnAssignedRoles(),wdCo
ntext.currentContextElement().getURoleSearch());
    //@@end
  }

  //@@begin javadoc:UnAssignRolesToUser()
  /** Declared method. */
  //@@end
  public void UnAssignRolesToUser( )
  {
    //@@begin UnAssignRolesToUser()
   try {
          String selUser =wdContext.currentContextElement().getSelectedUser();
          IRoleFactory fact = UMFactory.getRoleFactory();
          String id
=UMFactory.getUserFactory().getUserByLogonID(selUser).getUniqueID();
          int size = wdContext.nodeAssignedRoles().size();
          for (int i = 0; i < size; i++)
                if
(wdContext.nodeAssignedRoles().getAssignedRolesElementAt(i).getIsSelected())

   fact.removeUserFromRole(id,wdContext.nodeAssignedRoles().getAssignedRolesElementAt
(i).getRoleid());
          messageManager.reportSuccess("UnAssigned Roles Successfully");
          getAssignedRoles(selUser);
          getUnAssignedRoles();

   WDCopyService.copyElements(wdContext.nodeAssignedRoles(),wdContext.nodeSRAssignedR
oles());

   WDCopyService.copyElements(wdContext.nodeUnAssignedRoles(),wdContext.nodeSRUnAssig
nedRoles());
          SearchAssignedRoles();
          SearchUnassignedRoles();
   } catch (UMException e) {
          messageManager.reportException(e.getMessage(), true);
   }
    //@@end
  }

  //@@begin javadoc:SelectUser()
  /** Declared method. */
  //@@end
  public void SelectUser( )
  {
    //@@begin SelectUser()
   try {
                wdContext.nodeAssignedRoles().invalidate();
                wdContext.nodeUnAssignedRoles().invalidate();
                wdContext.nodeSRAssignedRoles();
                wdContext.nodeSRUnAssignedRoles();
                wdContext.currentContextElement().setARoleSearch(null);
                wdContext.currentContextElement().setURoleSearch(null);
                String selId = wdContext.currentContextElement().getSelectedUser();
                getAssignedRoles(selId);
                getUnAssignedRoles();
```

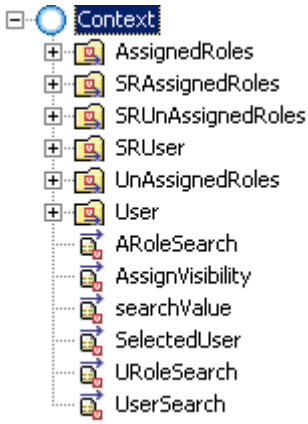
```
                assignedUsers = null;

   wdContext.currentContextElement().setAssignVisibility(WDVisibility.VISIBLE);

   WDCopyService.copyElements(wdContext.nodeUnAssignedRoles(),wdContext.nodeSRUnAssig
nedRoles());

   WDCopyService.copyElements(wdContext.nodeAssignedRoles(),wdContext.nodeSRAssignedR
oles());
         } catch (UMException e) {
                messageManager.reportException(e.getMessage(), false);
         }
    //@@end
  }

  /*
   * The following code section can be used for any Java code that is
   * not to be visible to other controllers/views or that contains constructs
   * currently not supported directly by Web Dynpro (such as inner classes or
   * member variables etc.). </p>
   *
   * Note: The content of this section is in no way managed/controlled
   * by the Web Dynpro Designtime or the Web Dynpro Runtime.
   */
  //@@begin others
  IWDMessageManager messageManager=null;
  private HashSet assignedUsers = null;

  /*
   * filterTable method is to search within the context node.
   *
   */
  private void filterTable(IWDNode nodeOriginal,IWDNode nodeDuplicate,String
serCriteria){
   nodeOriginal.invalidate();
   if(serCriteria==null||serCriteria.trim().equals("")||serCriteria.trim().equals("*"
)){
      WDCopyService.copyElements(nodeDuplicate,nodeOriginal);
      return;
   }
   serCriteria=serCriteria.replace('*',' ').trim();
   serCriteria=serCriteria.toUpperCase();
   int size=nodeDuplicate.size();
   for(int i=0;i<size;i++){
         String stst="";
         Iterator it=nodeDuplicate.getNodeInfo().iterateAttributes();
         while(it.hasNext()){
                stst=stst+"
"+nodeDuplicate.getElementAt(i).getAttributeValue(((IWDAttributeInfo)it.next()).getNa
me());
         }
       stst=stst.toUpperCase();
         if(stst.indexOf(serCriteria)!=-1){
         IWDNodeElement ele=nodeOriginal.createElement();
         it=nodeDuplicate.getNodeInfo().iterateAttributes();
         while(it.hasNext()){
```

```
                String name=((IWDAttributeInfo)it.next()).getName();

   ele.setAttributeValue(name,nodeDuplicate.getElementAt(i).getAttributeValue(name));
          }
          nodeOriginal.addElement(ele);
                }
   }
  }
/*
 * getAssignedRoles() method is to get the available roles for the selected user
 *
 */

  private void getAssignedRoles(String userID) throws UMException {
     wdContext.nodeAssignedRoles().invalidate();
     assignedUsers = new HashSet();
     IUserFactory uf = UMFactory.getUserFactory();
     IUser acc = uf.getUserByLogonID(userID);
     if (acc != null) {
       Iterator itr = acc.getRoles(true);
          IRoleFactory roleF = UMFactory.getRoleFactory();
          while (itr.hasNext()) {
            String id = (String) itr.next();
            IRole role = roleF.getRole(id);
            assignedUsers.add(id);
            IPublicRoleApp.IAssignedRolesElement ele =
wdContext.createAssignedRolesElement();
            ele.setRoleid(role.getUniqueID());
            ele.setRolename(role.getDisplayName());
            ele.setDescription(role.getDescription());
            wdContext.nodeAssignedRoles().addElement(ele);
          }
     }
  }
/*
 * getUnAssignedRoled() method is to get the Unassigned roles available in the
portal.
 *
 */
  private void getUnAssignedRoles() throws UMException {
     wdContext.nodeUnAssignedRoles().invalidate();
     IRoleFactory rolef = UMFactory.getRoleFactory();
     IRoleSearchFilter searchfilterrole = rolef.getRoleSearchFilter();
     ISearchResult searchResult = rolef.searchRoles(searchfilterrole);
     while (searchResult.hasNext()) {
        String unq = (String) searchResult.next();
           if (!assignedUsers.contains(unq)) {
              IPublicRoleApp.IUnAssignedRolesElement myelement
=wdContext.createUnAssignedRolesElement();
              IRole role1 = rolef.getRole(unq);
                 myelement.setRolename(role1.getDisplayName());
                 myelement.setRoleid(role1.getUniqueID());
                 myelement.setDescription(role1.getDescription());
                 wdContext.nodeUnAssignedRoles().addElement(myelement);
          }
     }
```

```
    }

    //@@end
  }
```

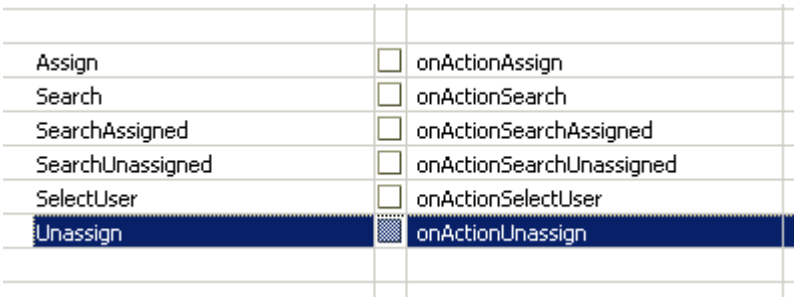
View Design

Map the entire context from Component controller to View. Look like this:



Define the actions as shown below.

- Search Action is to search the portal users.
- SearchAssigned and SearchUnAssigned to Search in the table
- Assign and Unassign to Assign and Un Assign the Roles to Portal Users
- SelectUser defines what to be done (retrieve the roles) after select the user.

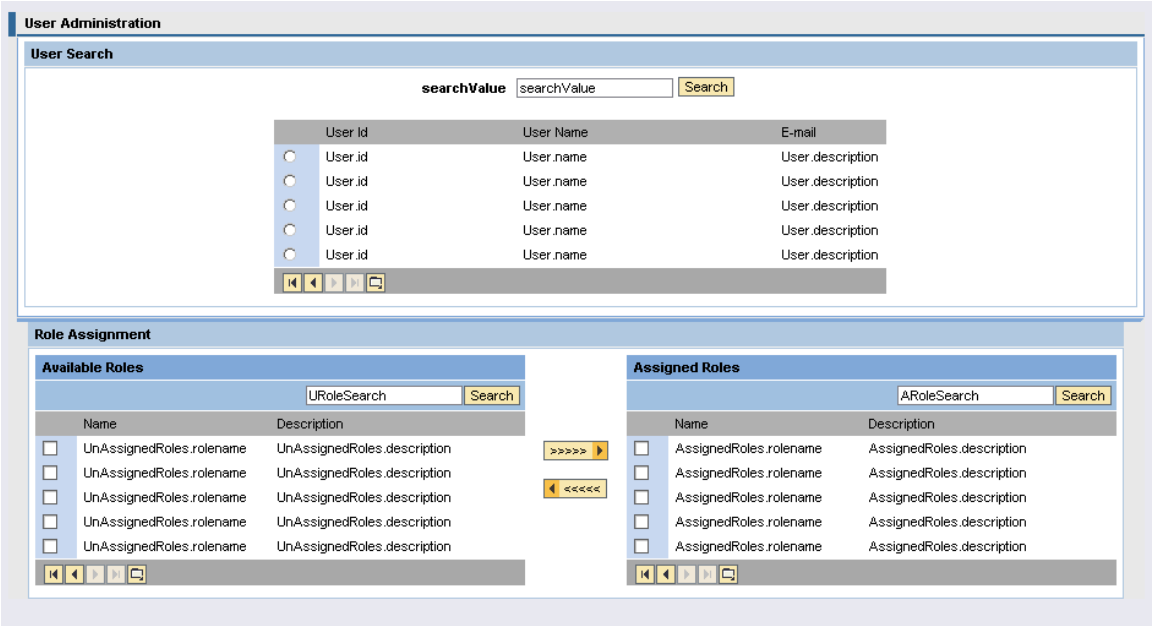


Design the Layout

The Layout is shown in below contains two panels, one is user search panel and another one is role operations panel. We will keep user search panel in Tray container UI Element and Operational panel in Group container UI Element.

User search panel contains one table and Operational panel contains two tables.

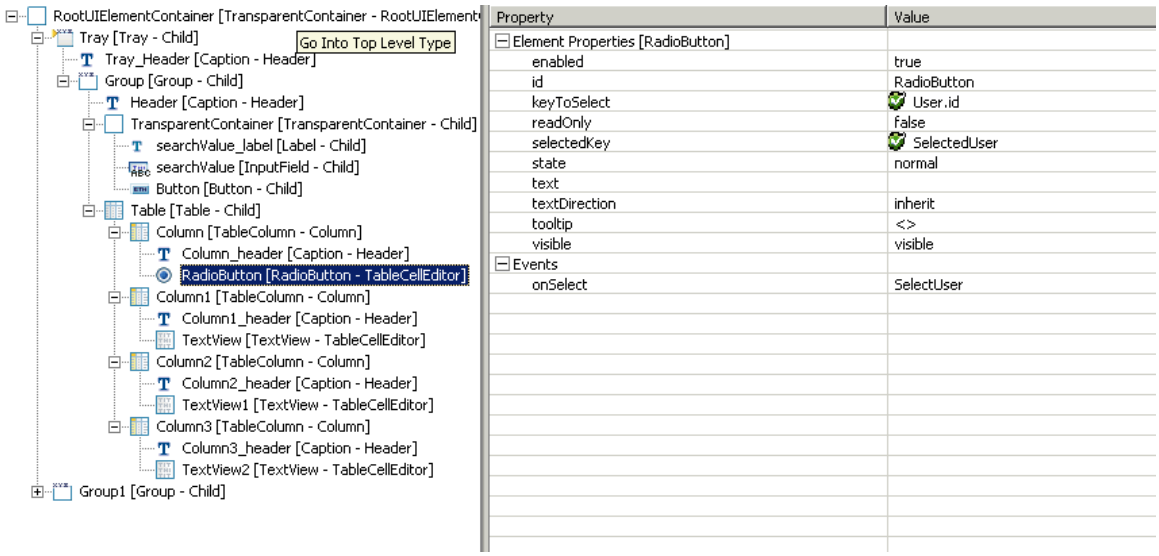
The Layout of RootContainerUIElement is GridLayout.



Design of the UserSearch Panel

- Insert a tray under RootUIElementContainer(tray layout: gridlayout,width:100%)
  - Insert a group under tray(group layout : gridlayout,design=sapcolor, width=100%, grid data :halign=center)
    - Insert a transparent container under Group(layout: Matrix layout, halign=center)
      - Insert label(text=SearchValue)
      - Insert InputField (value=bind to context attribute ‘searchValue’)
      - Insert Button(text=Search, bind Action ‘Search’)
    - Insert Table (datasource=bind to context node ‘User’, width=500%,halign=center)
      - Insert column(width=25px,)
        - Insert cell editor radio button(keyToSelect=Bind to attribute id under node user i.e. User.id , selectedKey=bind to attribute ‘Selected User’)
      - Insert Column(width=150px)
        - Insert cell editor textView(text=’User.id’)
      - Insert Column(width=200px)
        - Insert cell editor textView(text=’User.name’)
      - Insert Column
        - Insert cell editor text View(text=’User.description’)

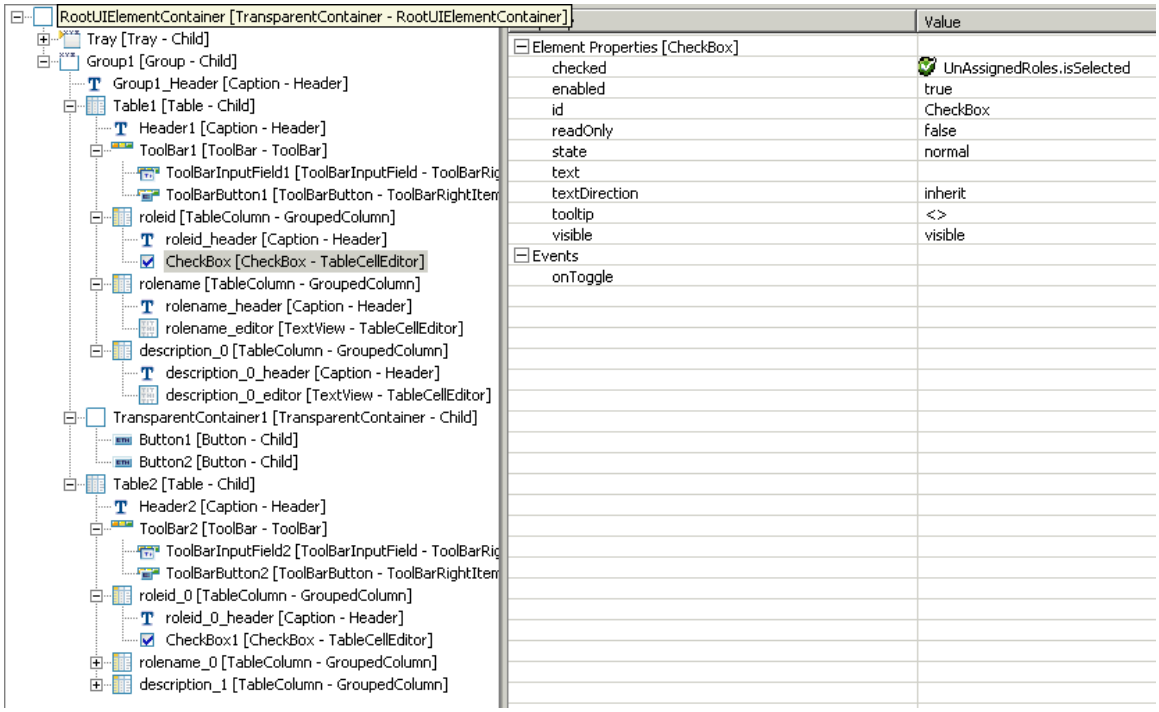
That looks like this:



Design of Operational Panel

- Insert a group under RootUIElementContainer (layout=gridlayout, colcount=3, halign=center, visibility='AssignVisibility').
  - Insert Table (datasource=bind to 'UnassignedRoles'.width=400px)
    - Insert toolbar
      - Insert toolbarrightItem type Inputfiled(value=bind attribute 'URoleSearch', onEnter=assign the action 'SearchUnassigned')
      - Insert toolbarrightItem type Button(text=Search, onAction=assign the action 'SearchUnassigned')
    - Insert Column(width=25px)
      - Insert cell editor as checkbox(checked='UnAssignedRoles.isSelected')
    - Insert Column(width=150px)
      - Insert cell editor as textView('text=UnAssignedRoles.rolename', tooltip='UnAssignedRoles.roleid')
    - Insert Column(width=200px)
      - Insert cell editor as textView('text=UnAssignedRoles.description', tooltip='UnAssignedRoles.roleid')
  - Insert TransparentContainer(layout =gridlayout, padding left= large, padding right=large)
    - Insert Button (design=next, ext=>>>>>,onAcion=Assign action 'Assign')
    - Insert Button (design=previous, text=<<<<<, onAcion=Assign action 'Unassign')
  - Insert Table (Data Source='AssignedRoled', width=400px)
    - Insert toolbar
      - Insert toolbarrightItem type Inputfiled(value=bind attribute 'ARoleSearch', onEnter=assign the action 'SearchAssigned')
      - Insert toolbarrightItem type Button(text=Search, onAction=assign the action 'SearchAssigned')
    - Insert Column(width=25px)
      - Insert cell editor as checkbox(checked=' AssignedRoles.isSelected')
    - Insert Column(width=150px)
      - Insert cell editor as textView(text='AssignedRoles.rolename', tooltip=' AssignedRoles.roleid')
    - Insert Column(width=200px)
      - Insert cell editor as textView(text=' AssignedRoles.description', tooltip=' AssignedRoles.roleid')

That looks like this:



View Implementation Code

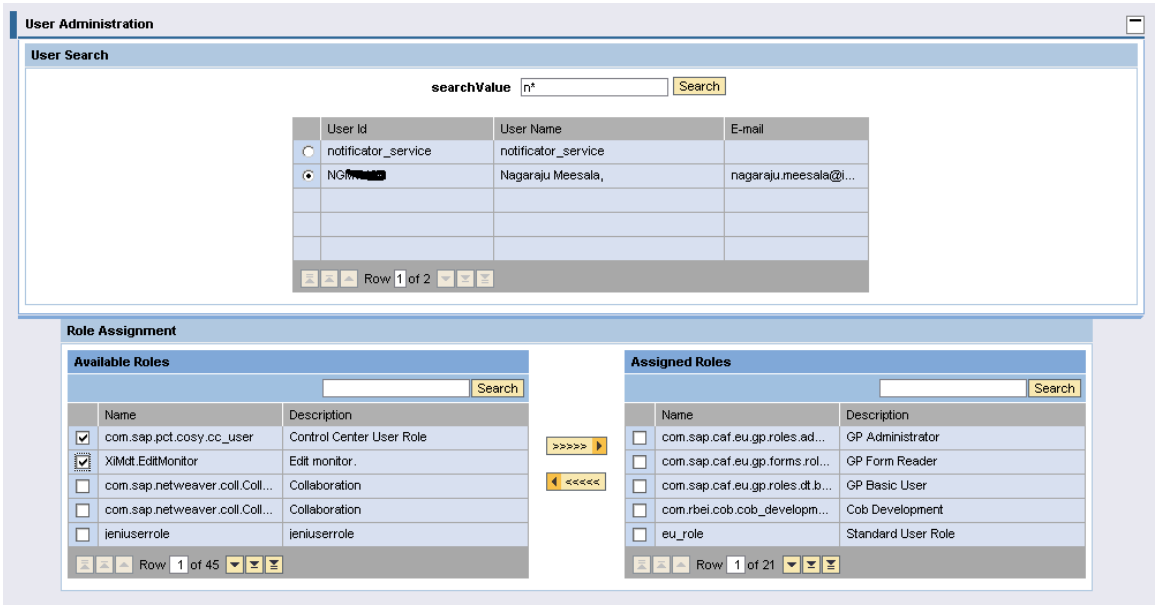
The Implementation of the View:

```
//@@begin imports
import com.rbei.bmh.ume.wdp.IPrivateRoleView;
import com.sap.tc.Web Dynpro.progmodel.api.WDCopyService;
import com.sap.tc.Web Dynpro.progmodel.api.WDVisibility;
//@@end

  public void wdDoInit()
  {
    //@@begin wdDoInit()

          wdThis.wdGetAPI().requestFocus(
                wdContext.currentContextElement(),
                wdContext.getNodeInfo().getAttribute("searchValue"));

   wdContext.currentContextElement().setAssignVisibility(WDVisibility.NONE);
    //@@end
  }


  public void onActionSearch(com.sap.tc.Web Dynpro.progmodel.api.IWDCustomEvent
wdEvent )
  {
    //@@begin onActionSearch(ServerEvent)
   wdThis.wdGetRoleAppController().SearchUser();
    //@@end
  }
  public void onActionSelectUser(com.sap.tc.Web Dynpro.progmodel.api.IWDCustomEvent
wdEvent )
  {
    //@@begin onActionSelectUser(ServerEvent)
   wdThis.wdGetRoleAppController().SelectUser();
    //@@end
  }


  public void onActionAssign(com.sap.tc.Web Dynpro.progmodel.api.IWDCustomEvent
wdEvent )
  {
    //@@begin onActionAssign(ServerEvent)
   wdThis.wdGetRoleAppController().AssignRolesToUser();
    //@@end
  }


  public void onActionUnassign(com.sap.tc.Web Dynpro.progmodel.api.IWDCustomEvent
wdEvent )
  {
    //@@begin onActionUnassign(ServerEvent)
          wdThis.wdGetRoleAppController().UnAssignRolesToUser();
    //@@end
```

```
  }

  public void onActionSearchAssigned(com.sap.tc.Web
Dynpro.progmodel.api.IWDCustomEvent wdEvent )
  {
    //@@begin onActionSearchAssigned(ServerEvent)
   wdThis.wdGetRoleAppController().SearchAssignedRoles();
    //@@end
  }

  public void onActionSearchUnassigned(com.sap.tc.Web
Dynpro.progmodel.api.IWDCustomEvent wdEvent )
  {
    //@@begin onActionSearchUnassigned(ServerEvent)
   wdThis.wdGetRoleAppController().SearchUnassignedRoles();
    //@@end
  }
```

Deploy

After you deploy and run the application, the output likes this:



Related Content

Creating First Web Dynpro Application

Online Tutorials and Samples of Web Dynpro Java

Working with tables in Web Dynpro

For more information, visit the User Interface Technology 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.