

How to Create a Knowledge Management (KM) Repository Service using KM API.

Applies to:

SAP Net Weaver Developer Studio (NWDS) version 7.0.11 and SAP Web Application Server (WAS) version 7.0.

Summary

Knowledge Management permits common access to unstructured information and documents in a distributed storage landscape. The repositories are directly linked to the portal with preconfigured or programmable connectors.

Knowledge Management provides various repository managers for different types of data and data sources. For example, data can be stored in a file system, or made available through a WebDAV server. To be able to use Knowledge Management functions on this data, you must configure a suitable repository manager for each data source.

Using this tutorial you can create a KM Repository Service using Knowledge Management API .

Author(s): Syed Rahamat Ali

Company: HCL Technologies Ltd, Kolkata

Created on: 08 November 2007

Author Bio



Syed Rahamat Ali is working in **HCL Technologies Ltd, Kolkata** as a **NetWeaver Consultant**.

Table of Contents

Applies to:	1
Summary.....	1
Author Bio	1
Abbreviation used	2
Pre-Requisites	2
Knowledge Requirements:.....	2
System Requirements:.....	2
Implementation:	2
Related Content.....	15
Disclaimer and Liability Notice.....	16

Abbreviation used

- NWDS** → SAP Net Weaver Developer Studio
- WAS** → SAP Web Application Server
- KM** → Knowledge Management

Pre-Requisites

Knowledge Requirements:

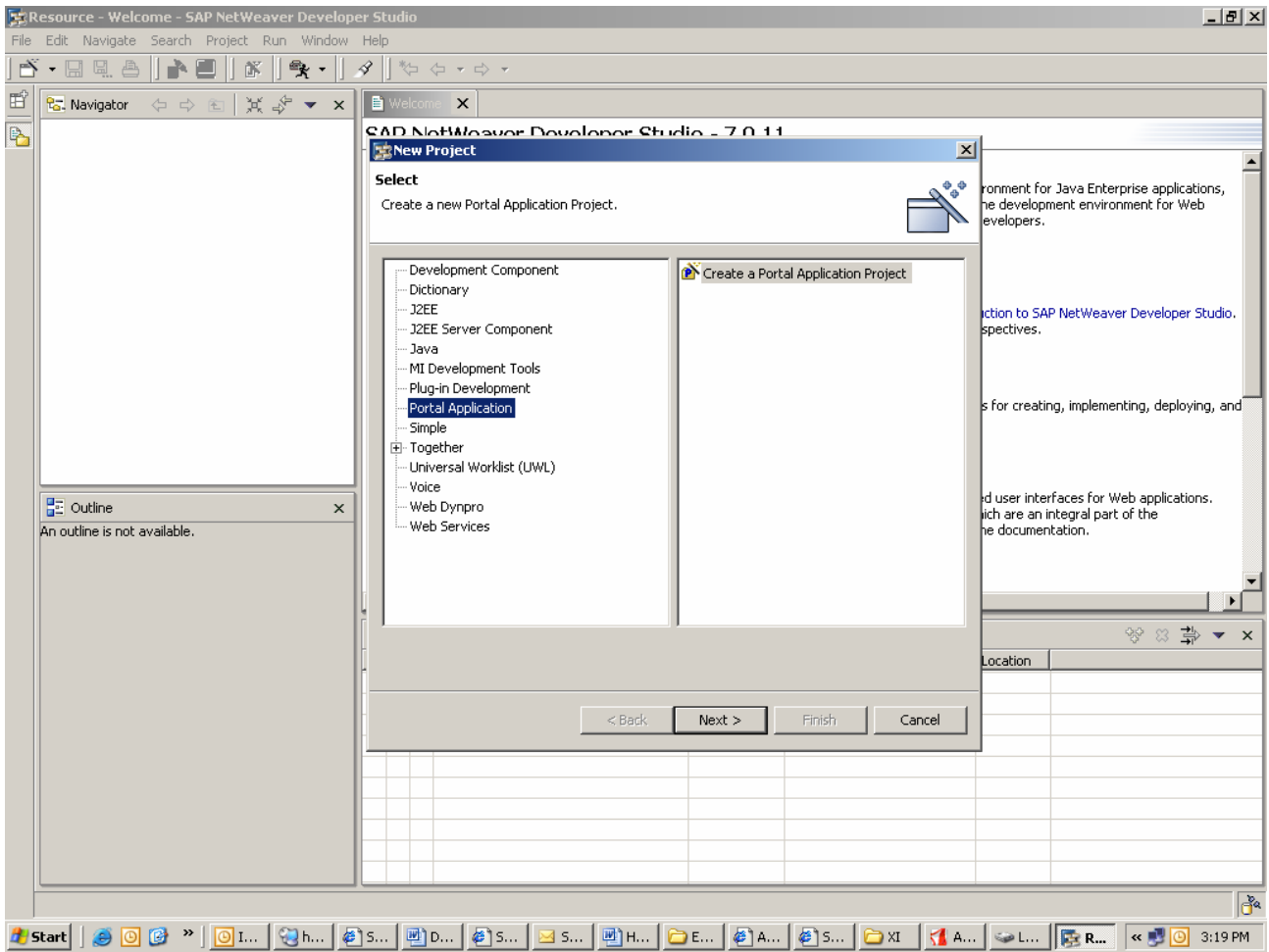
You should have good knowledge in KM Repository Framework.

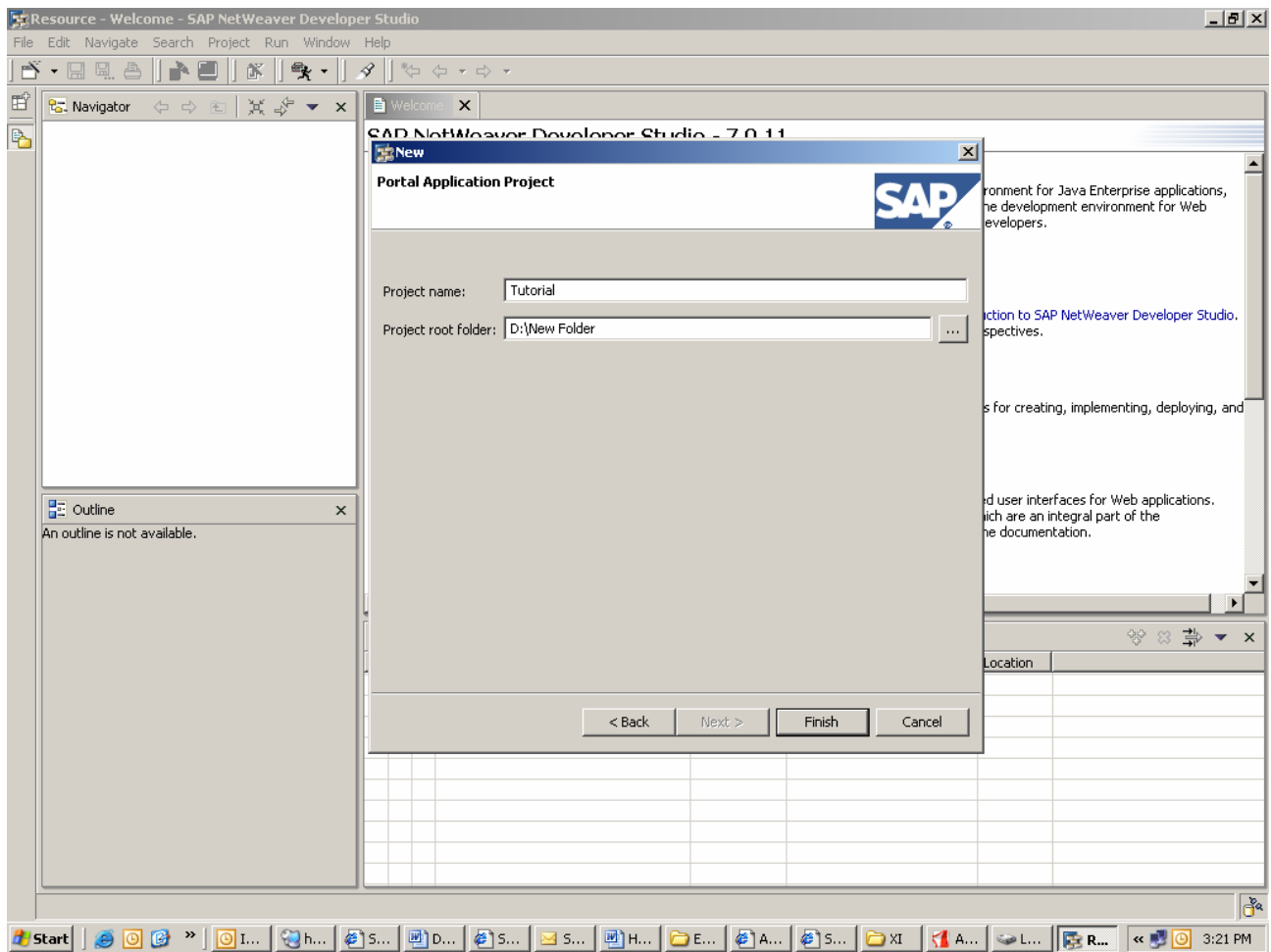
System Requirements:

- JDK 1.4.2_15,
- NWDS 7.0.11

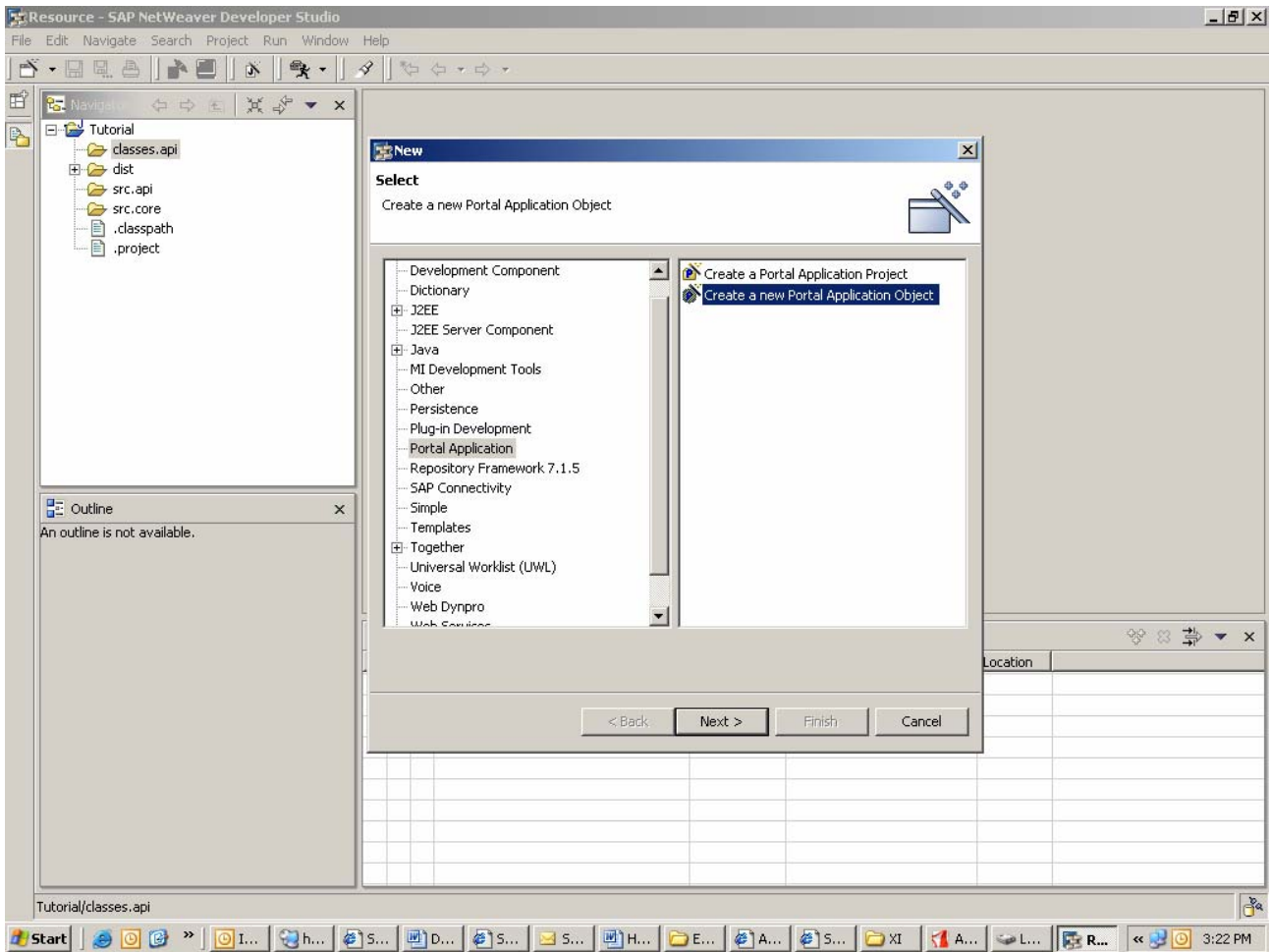
Implementation:

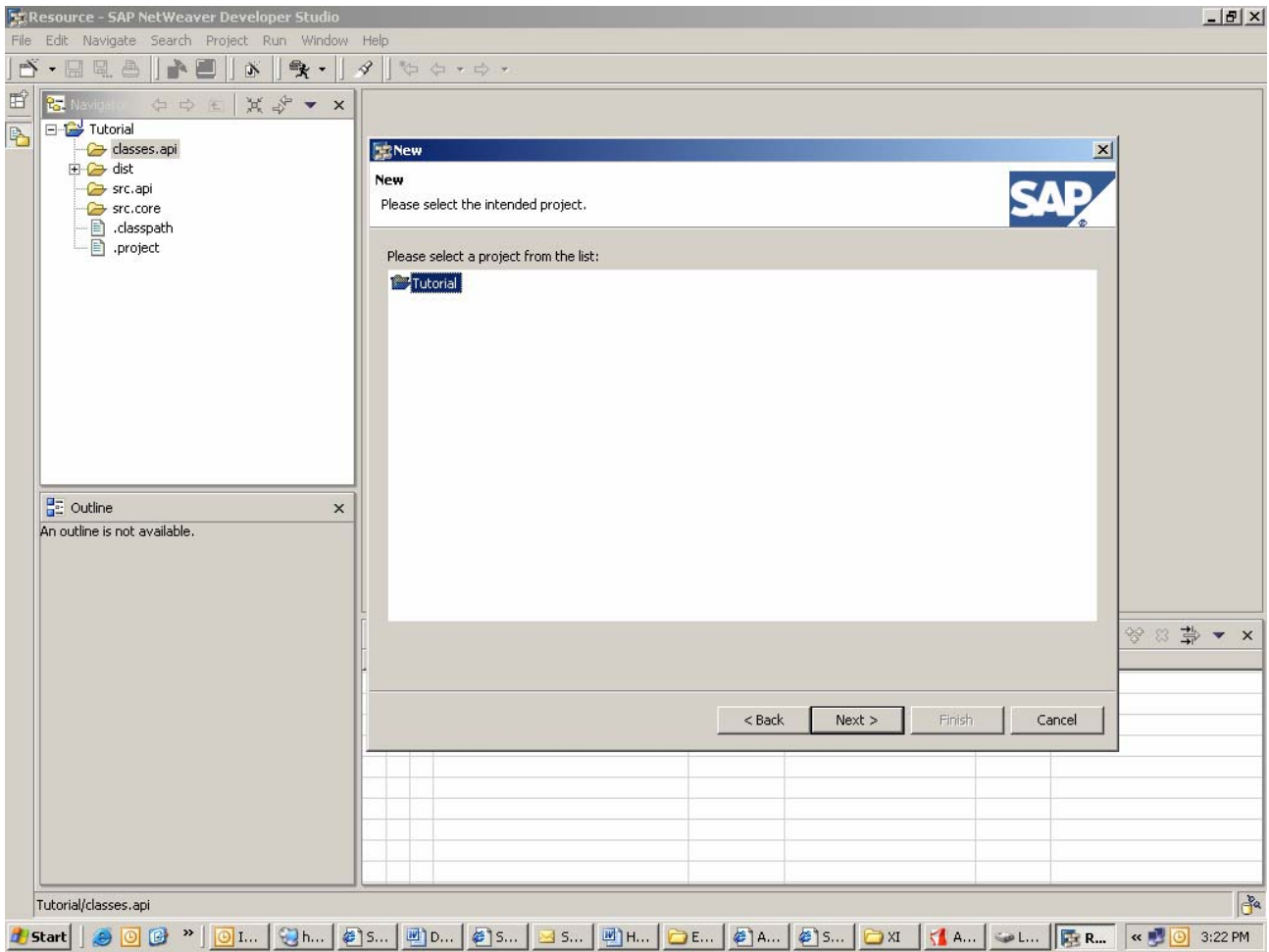
Step1: First Create a Portal Application workspace.

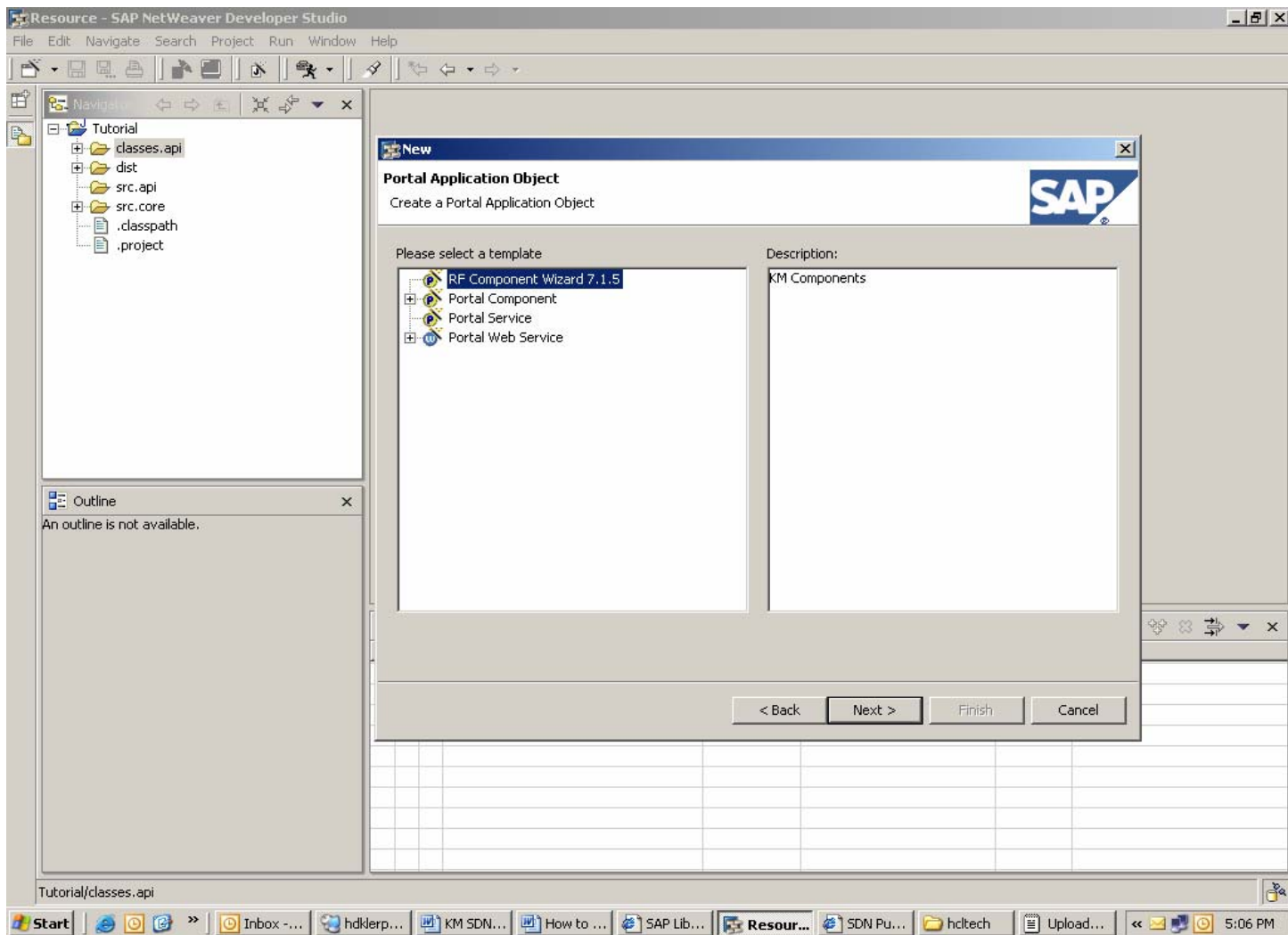




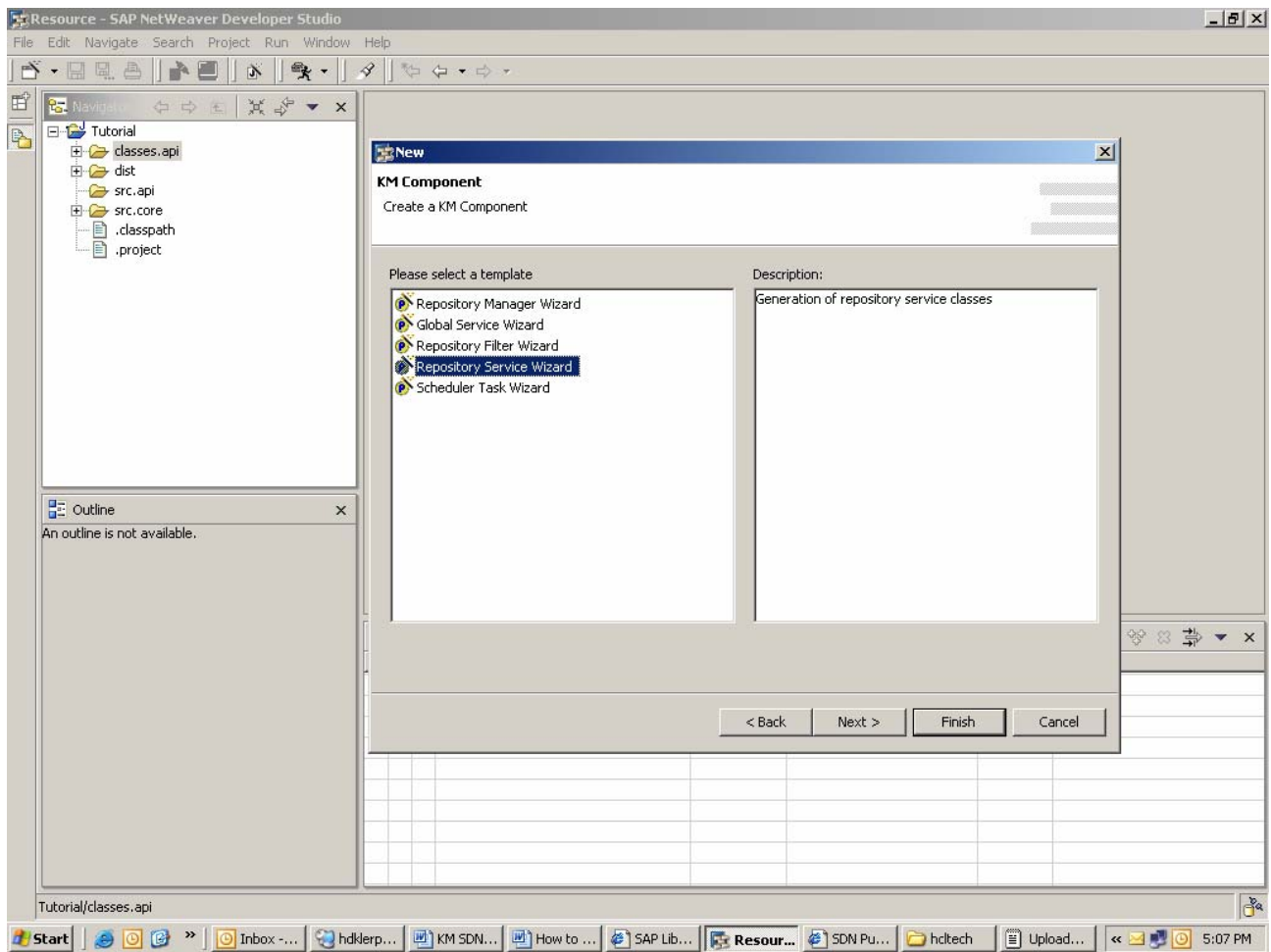
Step2: Then create a repository service using NWDS .



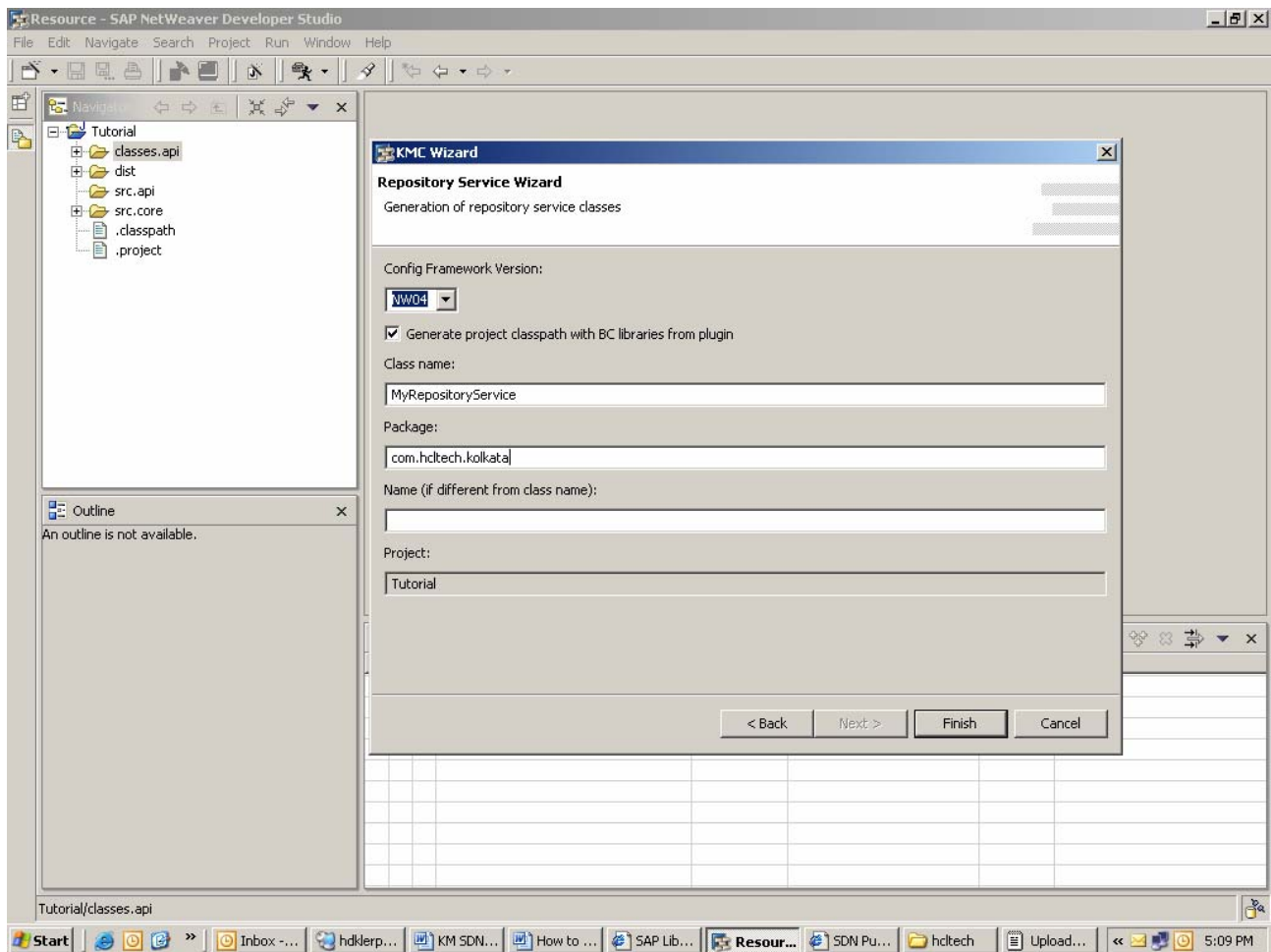




Select RF Component Wizard and click on next.



Select Repository Service Wizard and click next.



Specify a class name and package name and click on finish.

Step3: Open the IRFServiceWrapper.java file. The File is given below

```

package com.sap.netweaver.rf.wrapper;

import com.sapportals.portal.prt.service.IService;

public interface IRFServiceWrapper extends IService{
    public static final String KEY = "/* Mention the Fully qualified
                                   Repository Service name */";
}

```

Mention the KEY and save it.

Step4: Now Open the Repository Service file. The code of the file is given below.

```

package com.hcltech.kolkata;
import com.sapportals.wcm.repository.service.AbstractRepositoryService;
import com.sapportals.wcm.repository.service.ServiceNotAvailableException;

```

```

import com.sapportals.wcm.repository.manager.IResourceEventReceiver;
import com.sapportals.wcm.repository.manager.IRepositoryManager;
import com.sapportals.wcm.util.events.IEvent;
import com.sapportals.wcm.crt.component.*;
import com.sapportals.wcm.crt.configuration.*;
import com.sapportals.wcm.WcmException;
import java.util.Collection;

// implements IMyNewRepositoryService interface
/*
  Note: IReconfigurable and IResourceEventReceiver interfaces are optional
*/

public class MyRepositoryService extends AbstractRepositoryService implements
IReconfigurable, IResourceEventReceiver {

    private static final String TYPE = "MyRepositoryService";

    public MyRepositoryService() {
        super();
        // Do not add code here. Add it to startUpImpl() instead
    }

    public String getServiceType() {
        return MyRepositoryService.TYPE;
    }

    protected void startUpImpl(Collection repositoryManagers) throws
ConfigurationException, StartupException {
        // implement this method as follows:
        // - Verify configuration data
        // - Get references to other needed (global) services
        // - Check whether other repository services (this service depends on) are
also assigned to the repository managers
        // - Usually the service registers itself for certain events at all
repository managers
        //
        /*
        try {
        }
        catch (Exception e) {
            throw new StartupException(e.getMessage(), e);
        }
        */
    }

    protected void shutDownImpl() { }

    protected void addRepositoryAssignment(IRepositoryManager mgr) throws
ServiceNotAvailableException {
        // Implement this method: Usually the service registers itself for certain
events at the repository manager.
    }

    protected void removeRepositoryAssignment(IRepositoryManager mgr) throws
WcmException {

```

```

    // Implement this method: Usually the service must unregister itself as an
    event handler.
    }

    public void reconfigure(IConfiguration config) throws ConfigurationException {
        this.stateHandler.preReconfigure();
        // check the new configuration data
        /*
        try {
        }
        catch (ConfigurationException ex) {
            this.stateHandler.postReconfigure(ex);
            throw ex;
        }*/

        this.config = config;
        this.stateHandler.postReconfigure();
    }

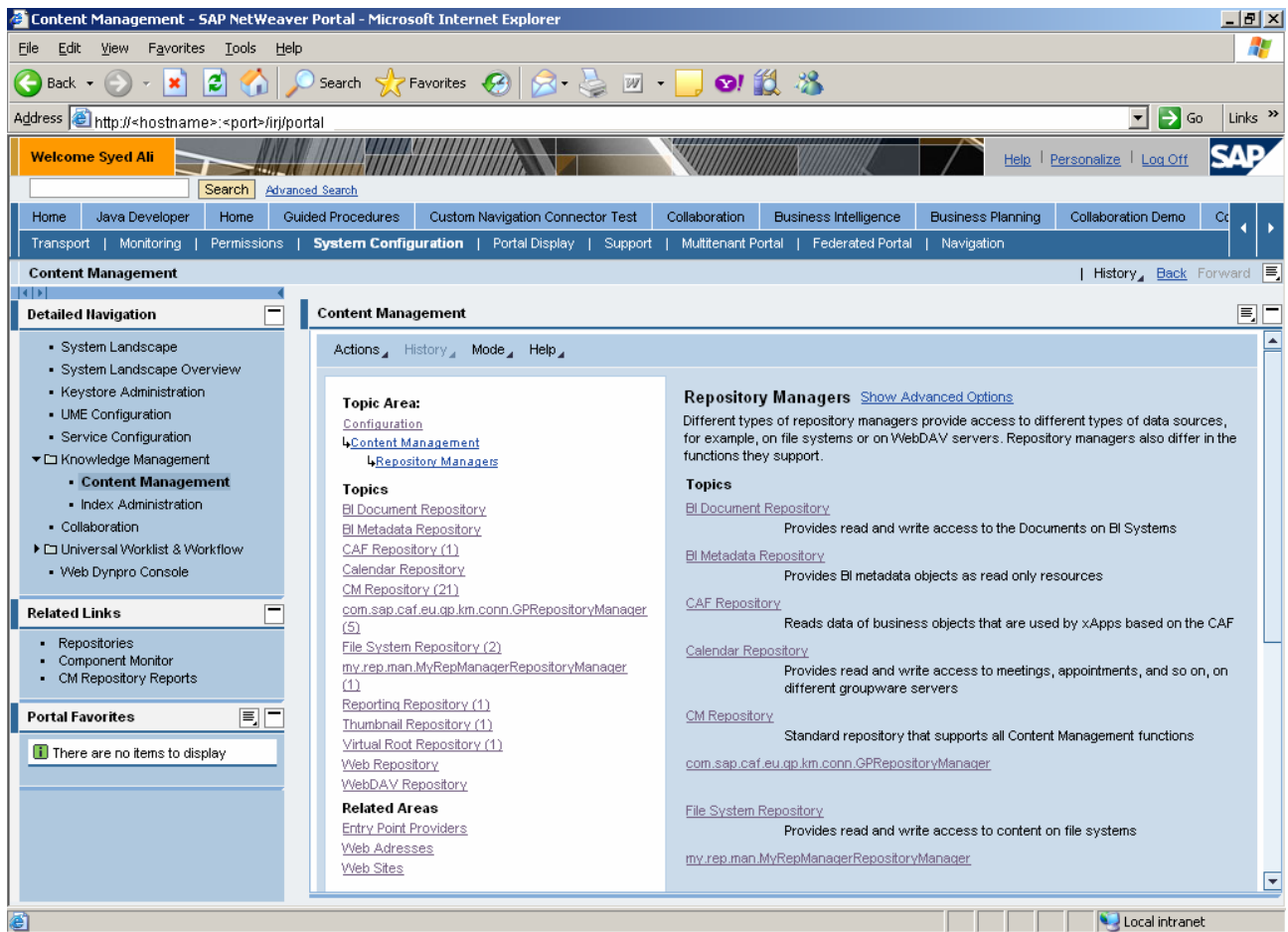
    public void received(IEvent event) {
    }
}

```

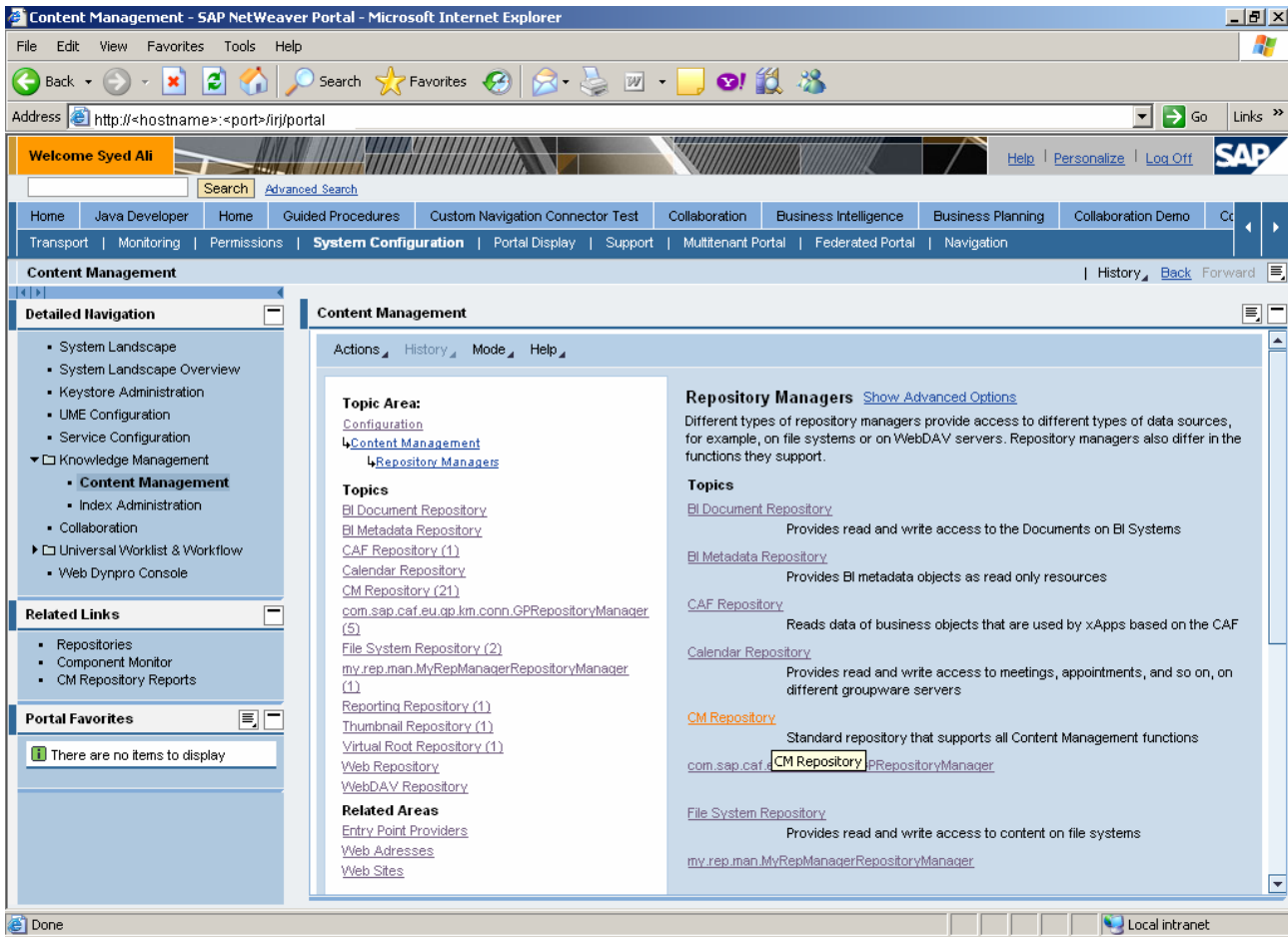
You just registered your event on `addRepositoryAssignment()` and Call this method from `startUpImpl()`. Then you unregistered all the event on `removeRepositoryAssignment()` which you have already registered.

Step5: Create the par file and upload it to WAS.

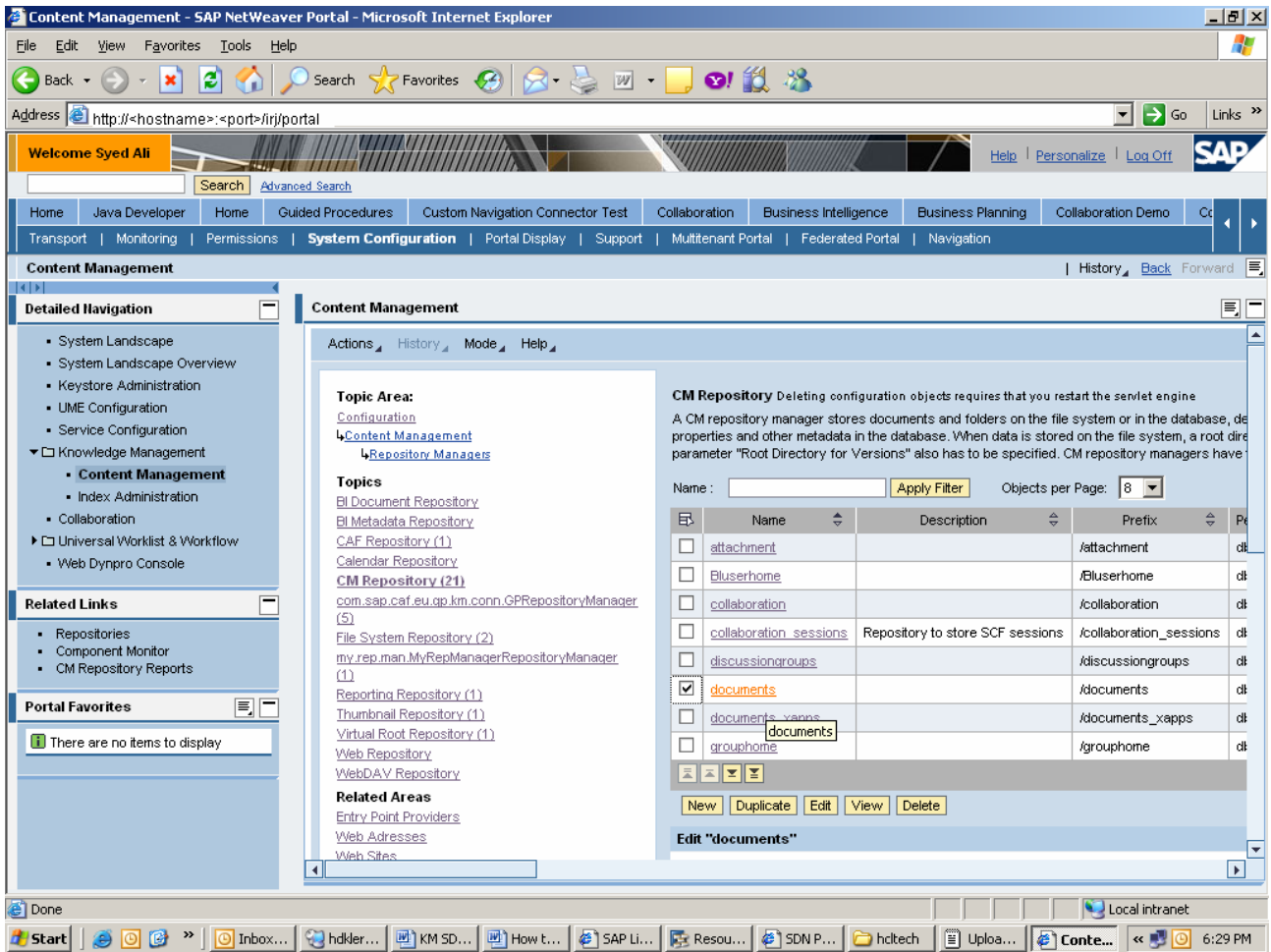
Step6: Open the portal console in browser. Click system Administrator -> System Configuration->Knowledge Management ->Content Management.



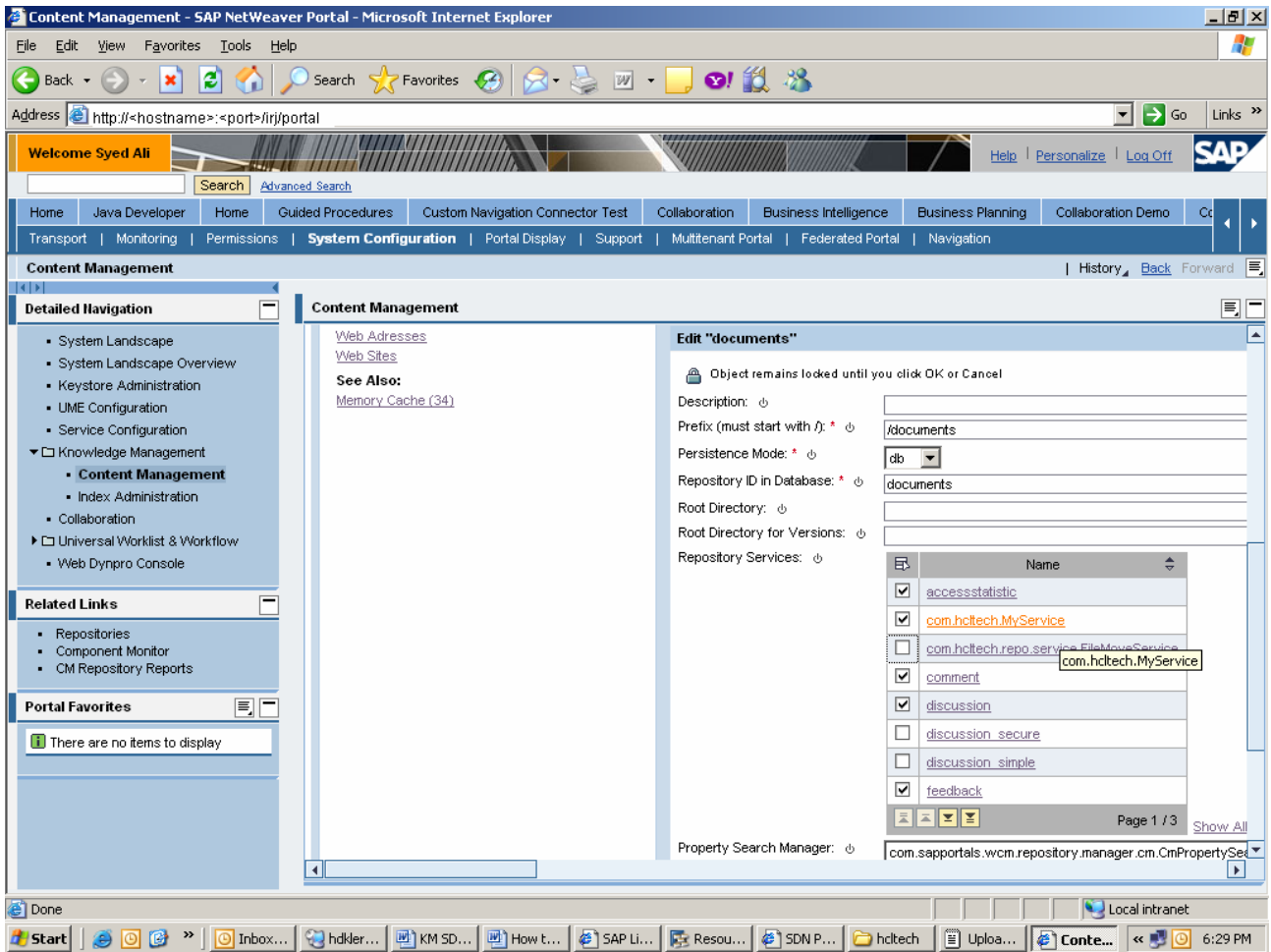
Step7: Click the Repository Manager. Now click the type of repository manager.(e.g. Click CM Repository)



Step8: Click the repository where you want to activate your service and then click edit button.



Step9: Select the repository and edit the repository and add your service to that repository to servlet engine.



Related Content

- [SDN Help](#)
- [SAP Help](#)

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.