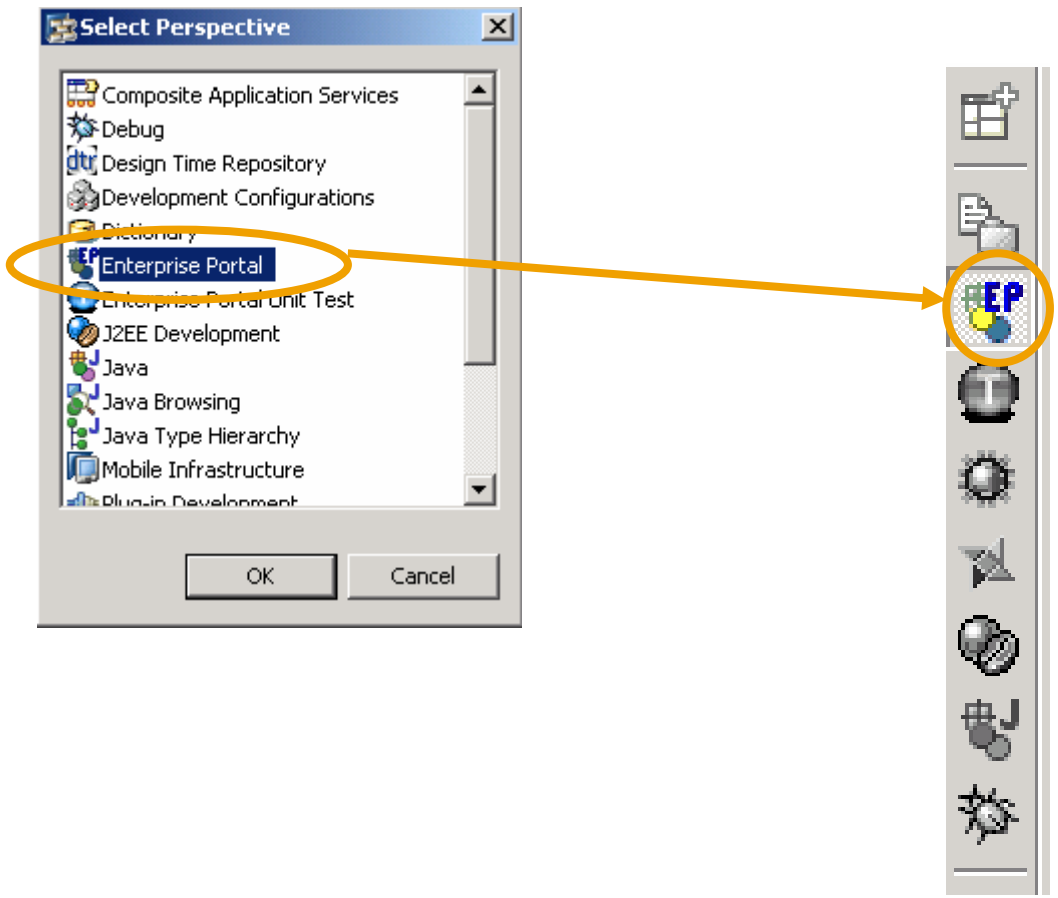
SAP NetWeaver Developer Studio for KMC

Requirements

- **Install a local JDK**
- **Install a local SAP NetWeaver Developer Studio**
  - **http://service.sap.com/instguidesNW04 -> Installation -> Installation Guide -> SAP NetWeaverDeveloper Studio)**
- **Remote or local SAP Enterprise Portal with KMC**

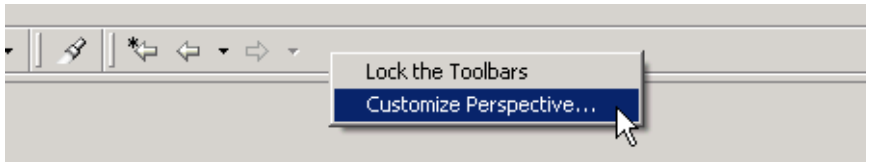
Activate the Enterprise Portal Perspective:

- Go to Window – Open Perspective – Other – Select Enterprise Portal

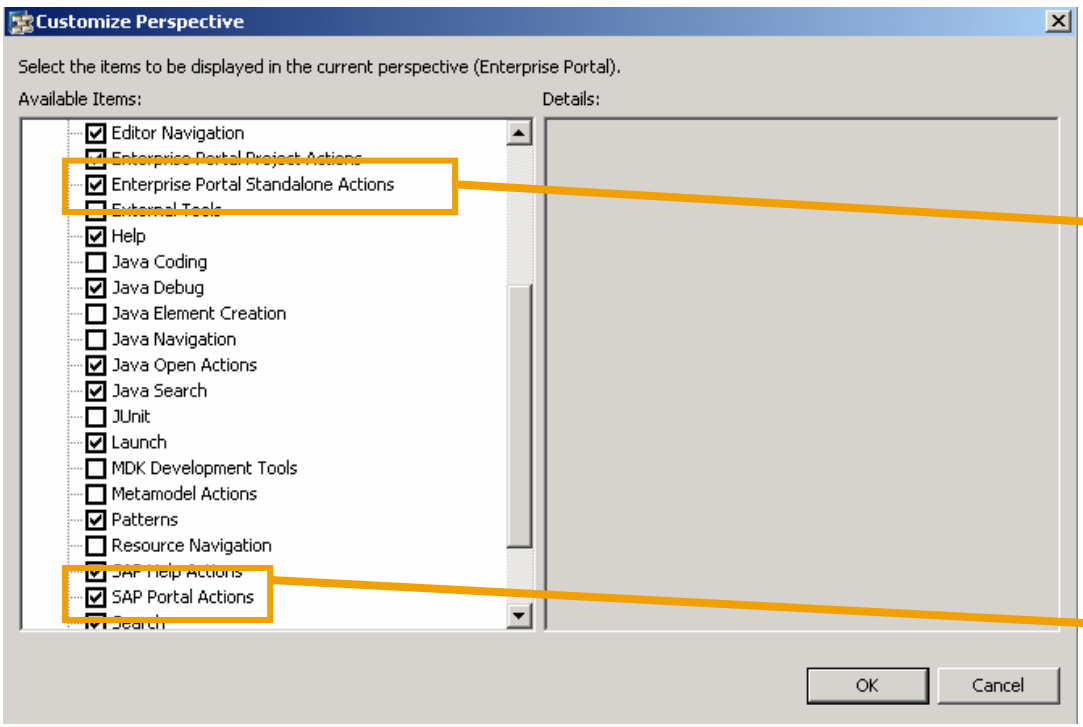


Customize the Toolbar to get the EP & KM PlugIn icons:

right-click the toolbar -> Customize Perspective

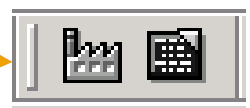


Select the items for the Portal and KM Plugins



EP Plugins

KM Plugins

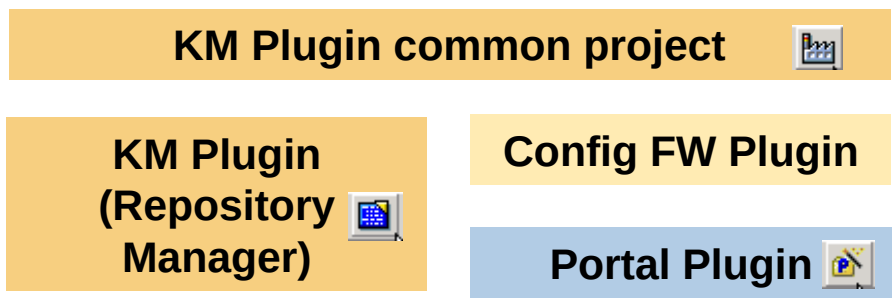


Portal-Plugins

- Creates an empty portal application project
  - Creates source development folders for API & Core coding
  - Creates a filesystem structure for PAR file creation
    - Public area for client content
    - Private area for server side content
  - Creates a deployment descriptor file (portalapp.xml)
- Config Framework plugin
- Support for Portal specific components (portal services, portal components)
- Support for WebServices

KM-Plugins

- Creates source code skeletons for CM components
- Creates configuration entries (XML files) for config framework
- Creates Wrapper classes for registration at CrtClassLoader

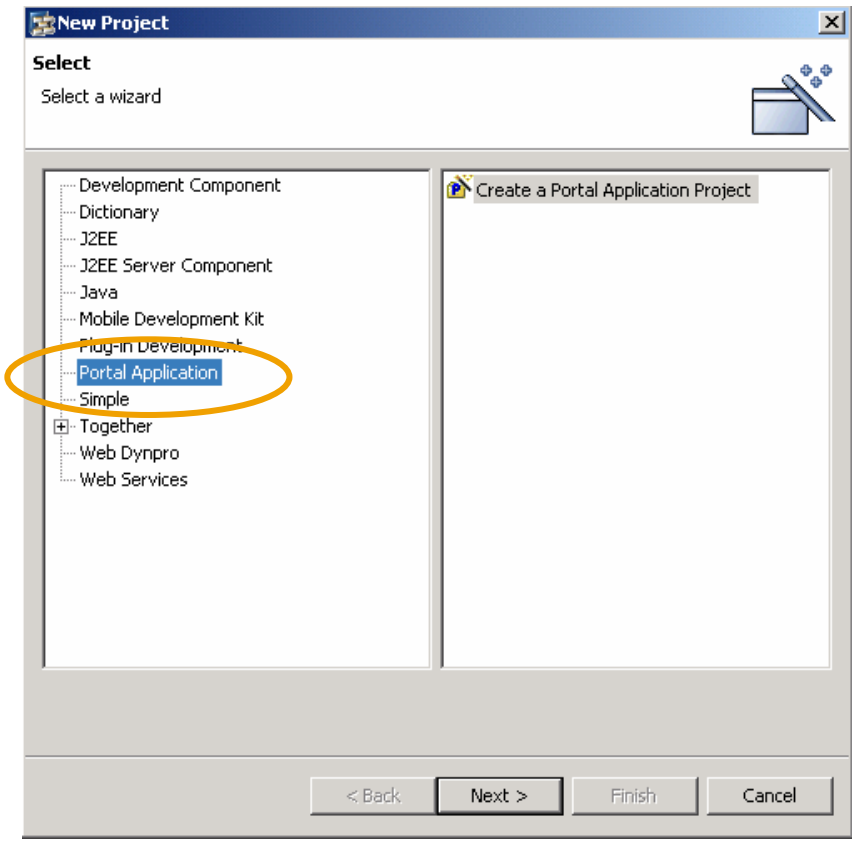


Steps:

1. **Create a new Portal Application Project**
2. **Integrate the KM jar files**
3. **Create a new Portal Application Object (KM component)**
4. **Select the Portal Application Project you want to create the KM component in.**
5. **Specify technical data**
   - **KM component ID**
   - **Java package name**
   - **Options used by KM component (optional)**
6. **Finalize project**

Create a new Portal Application project (1)

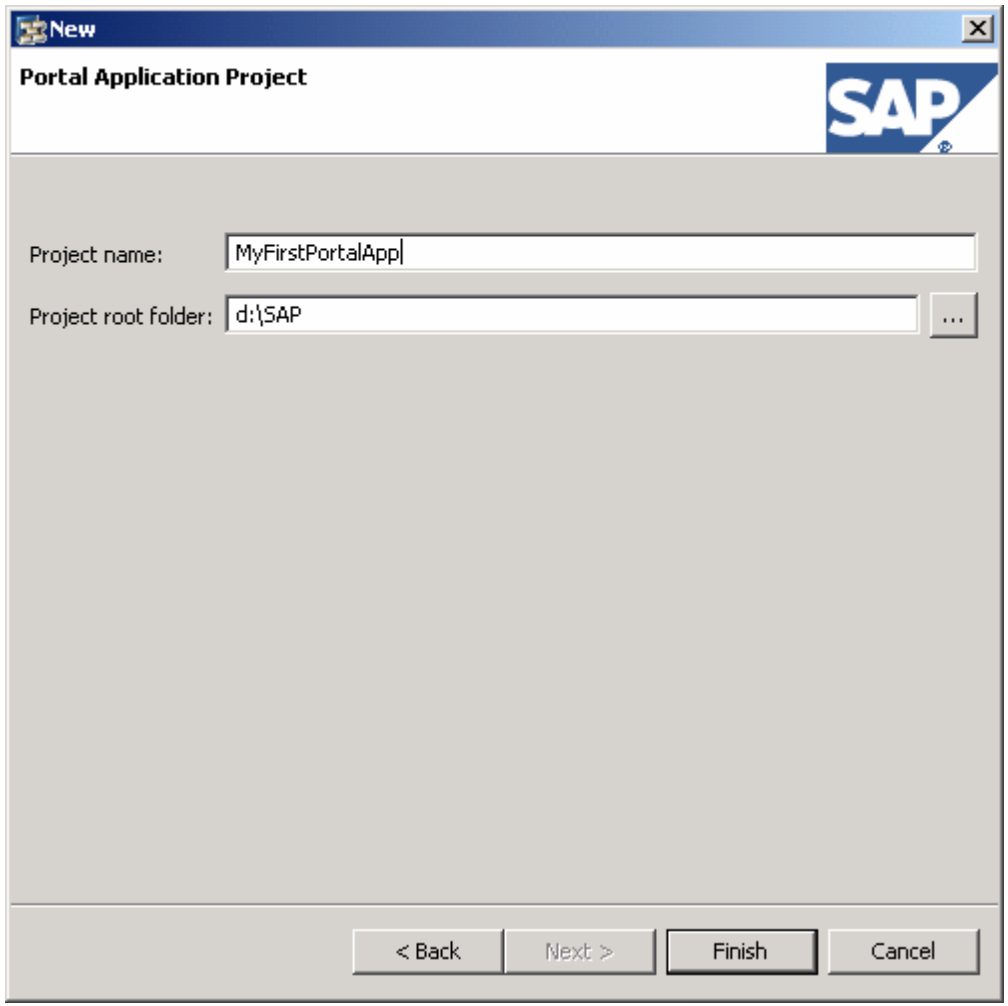
- From the menu, choose: File -> New ... -> Project



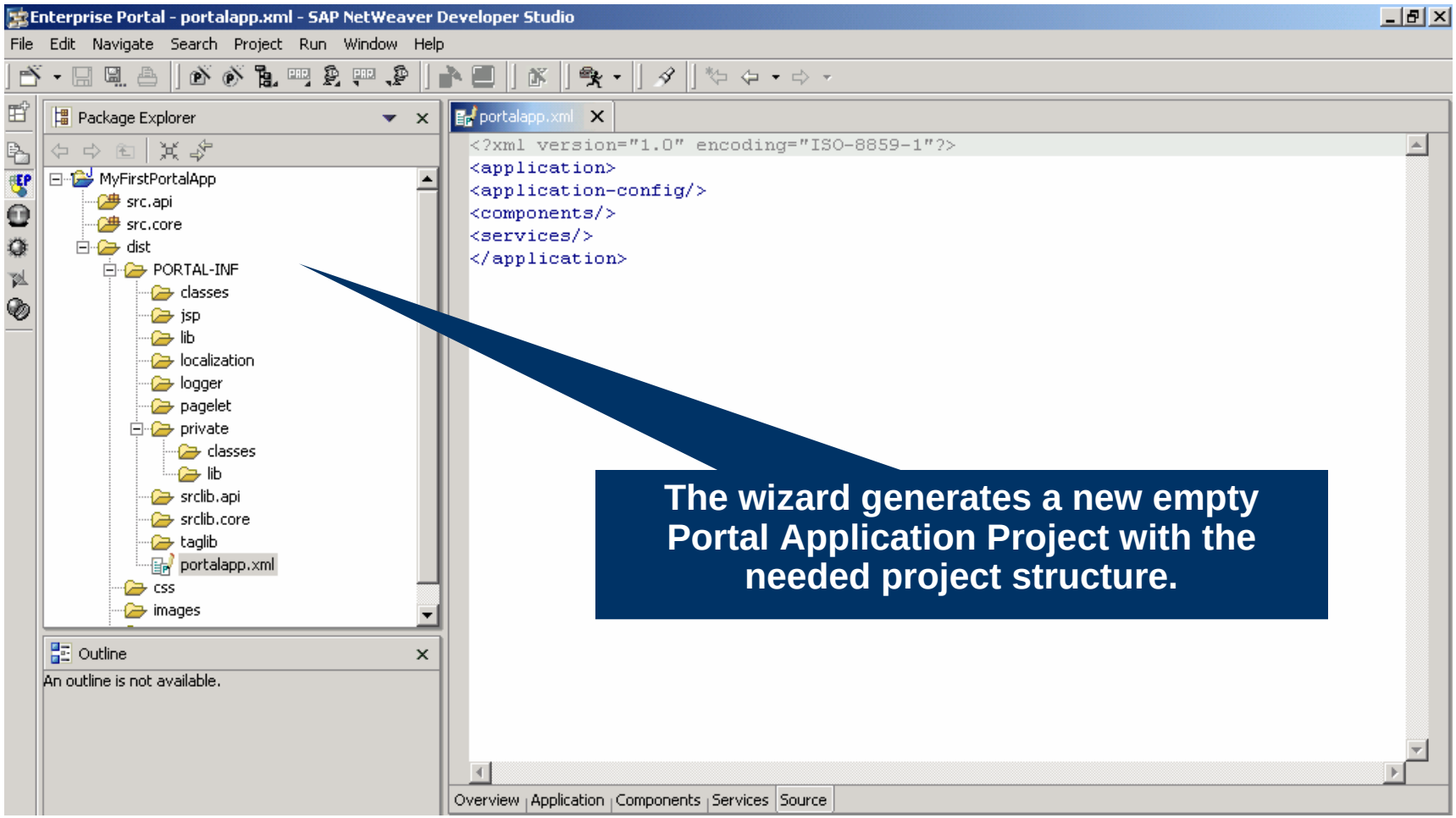
Or choose the appropriate icon in the toolbar:

Create a new Portal Application project (2)

Define a project name and the root folder for your project:



Create a new Portal Application project (3)



Portal Applications ...

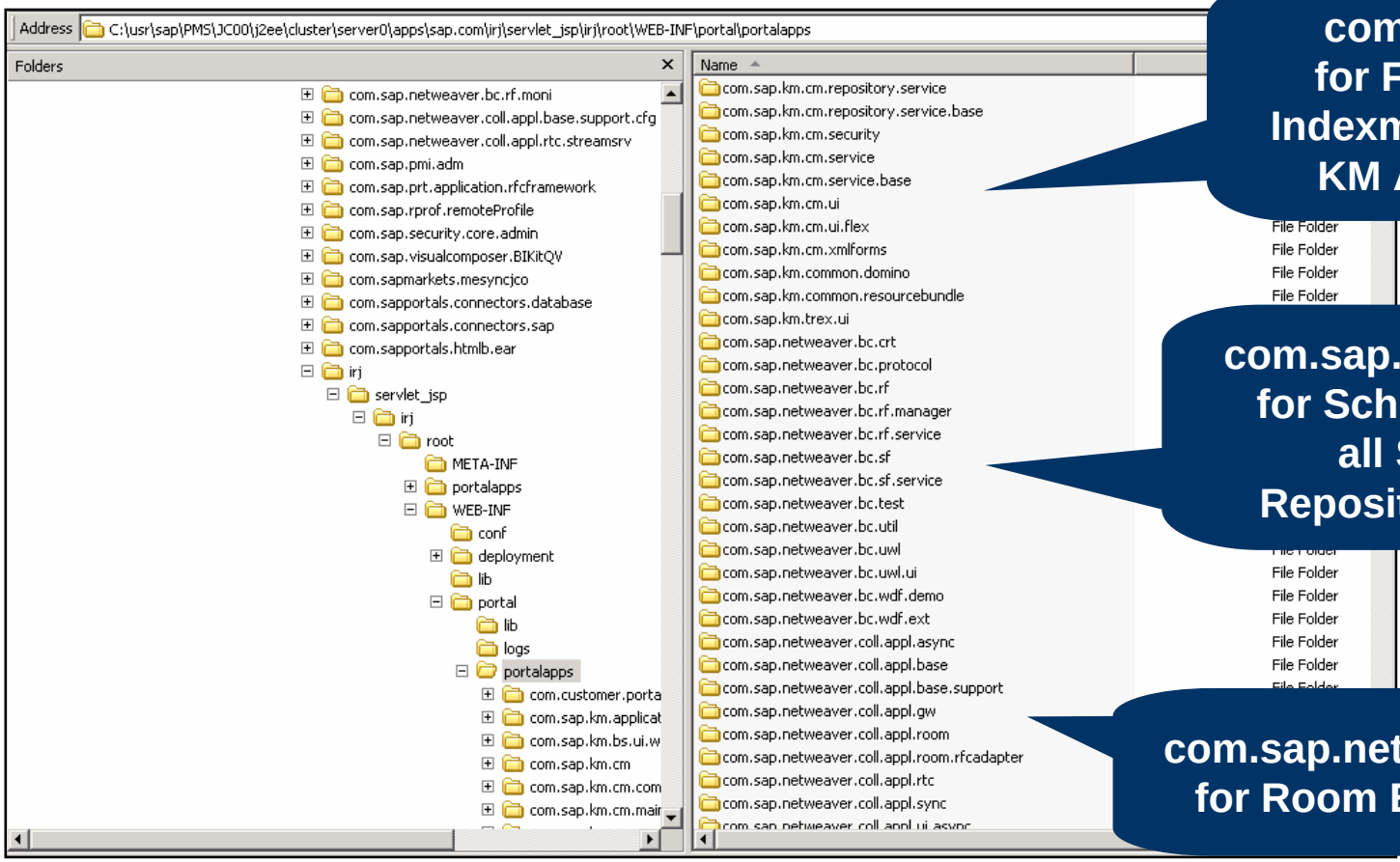
- **... are bundles of Portal Components and Portal Services.**
- **... are packaged as PAR (Portal Application ARchive) files – format that the Portal Runtime accepts for deployment of Portal Applications.**

- **A PAR file is an archive file/standard JAR file (ZIP format) containing a Portal Application with a XML-based Portal Application Deployment Descriptor (DD) called portalapp.xml.**
- **A DD describes the portal application itself and provides information required at runtime.**

SAP NetWeaver Developer Studio For KMC

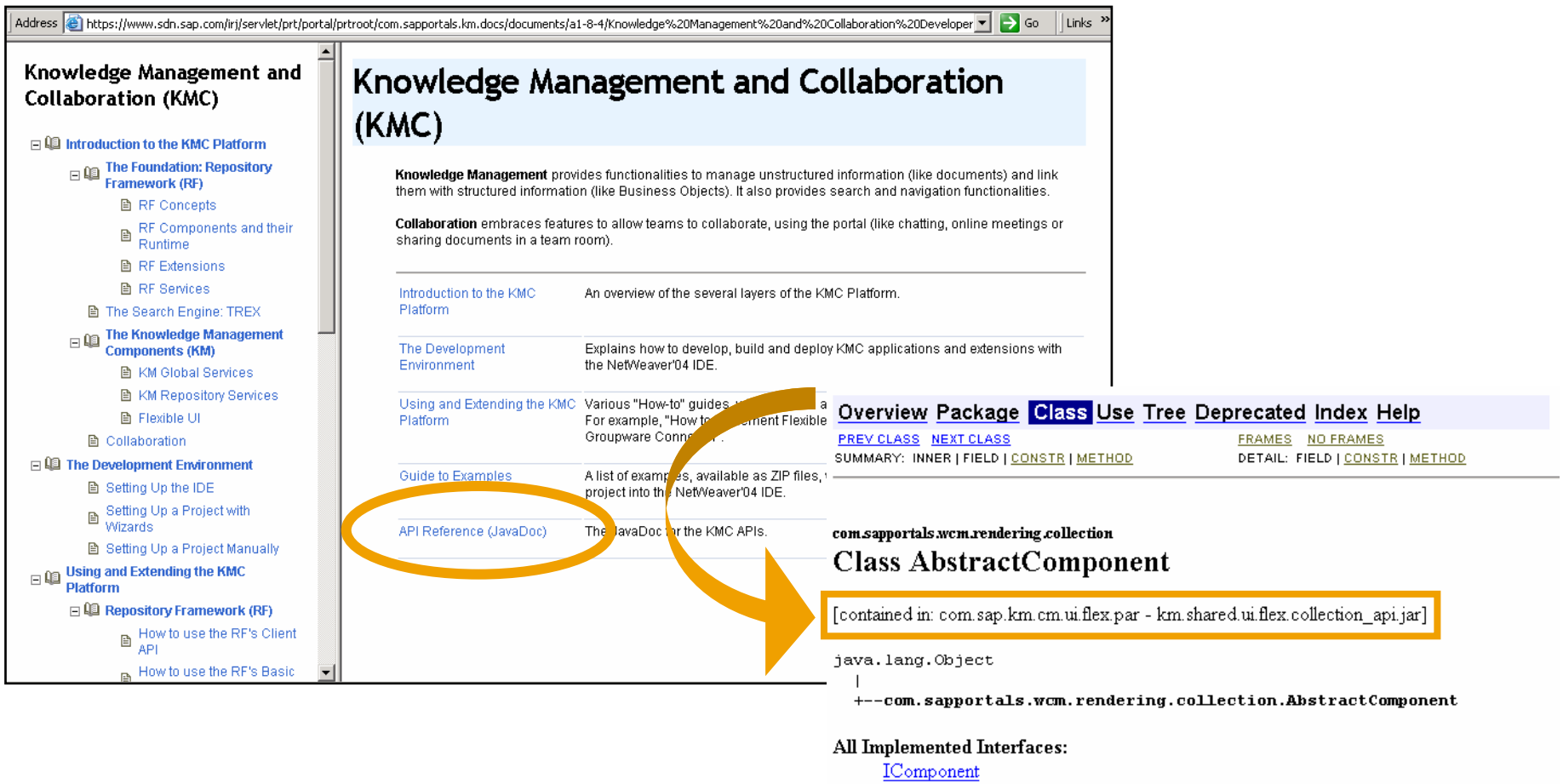
- **Portal Applications contain two types of resources:**
  - **Web Resources**
    - **Accessible via http(s) requests to a web server**
    - **All files NOT under PORTAL-INF (WEB-INF)**
  - **Non-Web Resources**
    - **Not accessible via an http(s) request**
    - **All files under PORTAL-INF (WEB-INF)**
- **A Portal Application can contain:**
  - **several Portal Components and several Portal Services**
  - **only 1-n Portal Components**
  - **only 1-n Portal Services**
- **Each Portal Component can be accessed by <par file name>.<component name>**

Integrate the KM libraries from your Portal installation: take the public JAR files from the par files

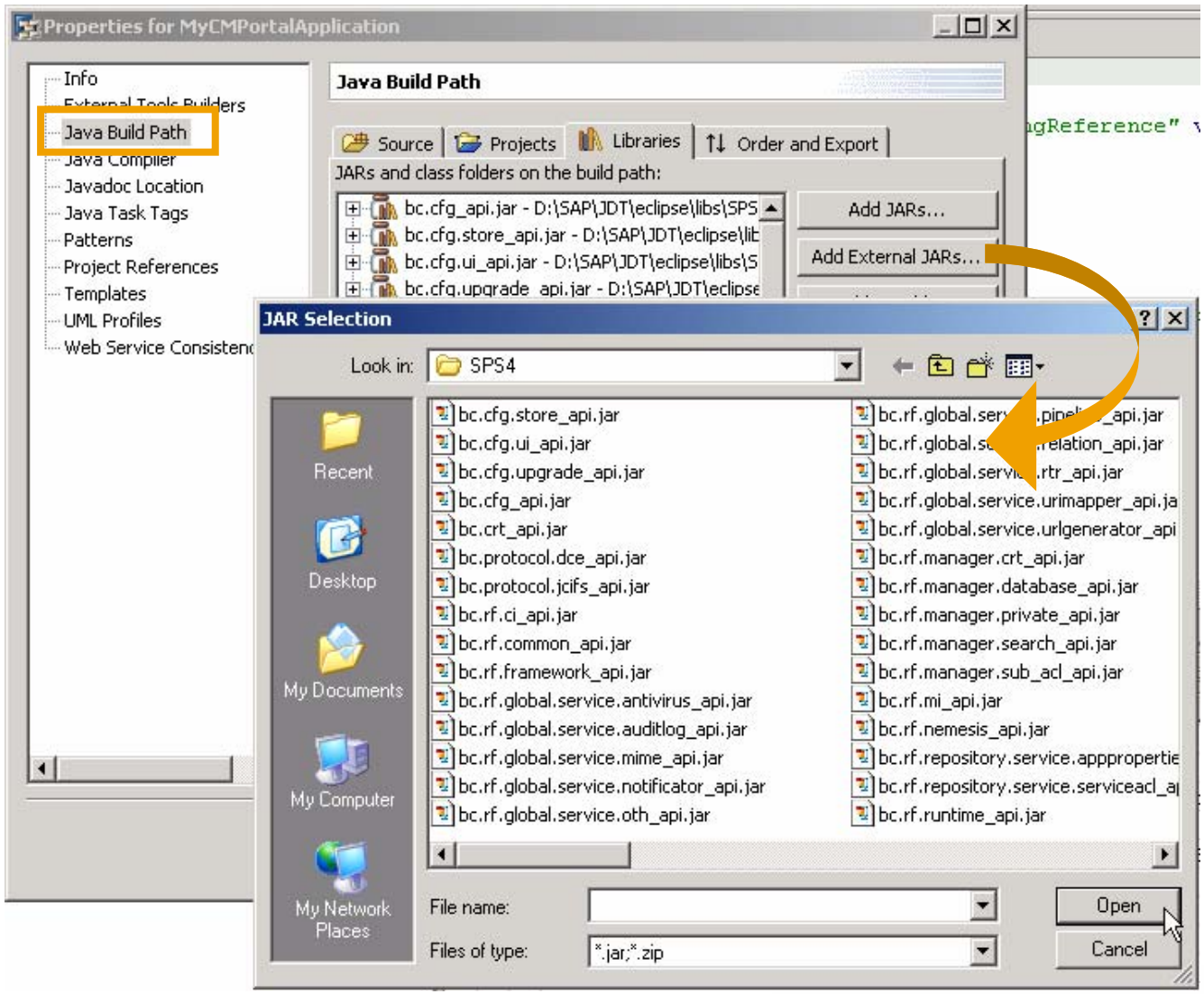


Hint: copy all needed jar files to a local folder

Go to the SDN – Knowledge Management – Quick Links:
Knowledge Management and Collaboration Developer's Guide

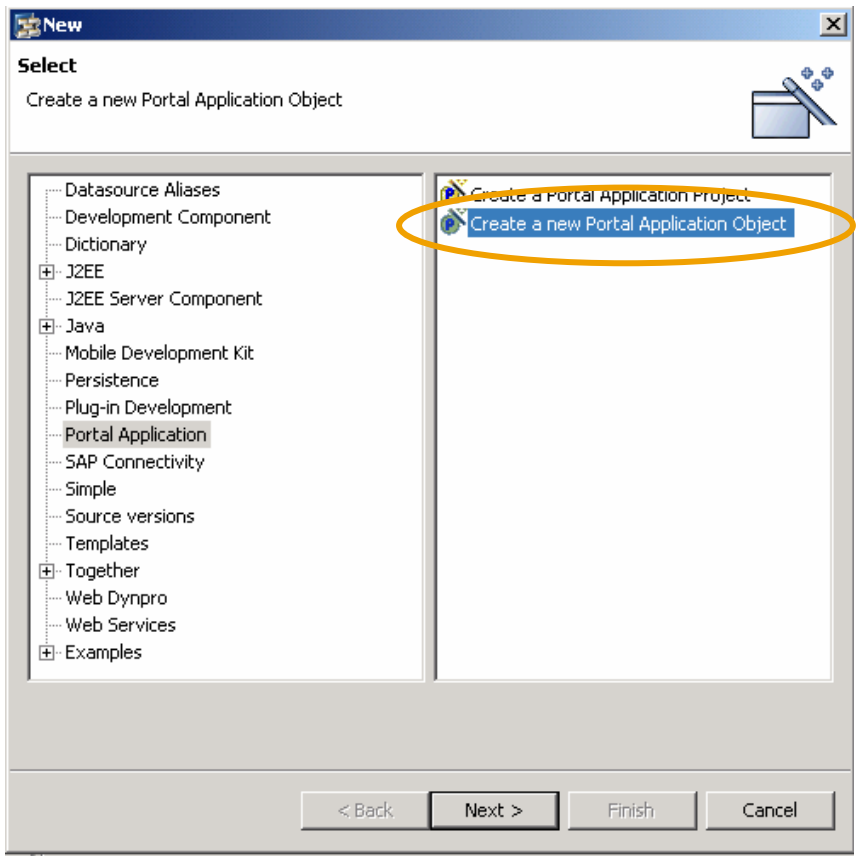


Add KM libraries to your Portal Application KM project



Create a new Portal Application Object (1)

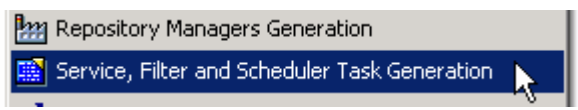
- From the menu, choose: File ➔ New ... ➔ Create a new Portal Application Object
- OR File ➔ New ... ➔ Other ➔ Create a new Portal Application Object



OR choose the appropriate icon in the toolbar:

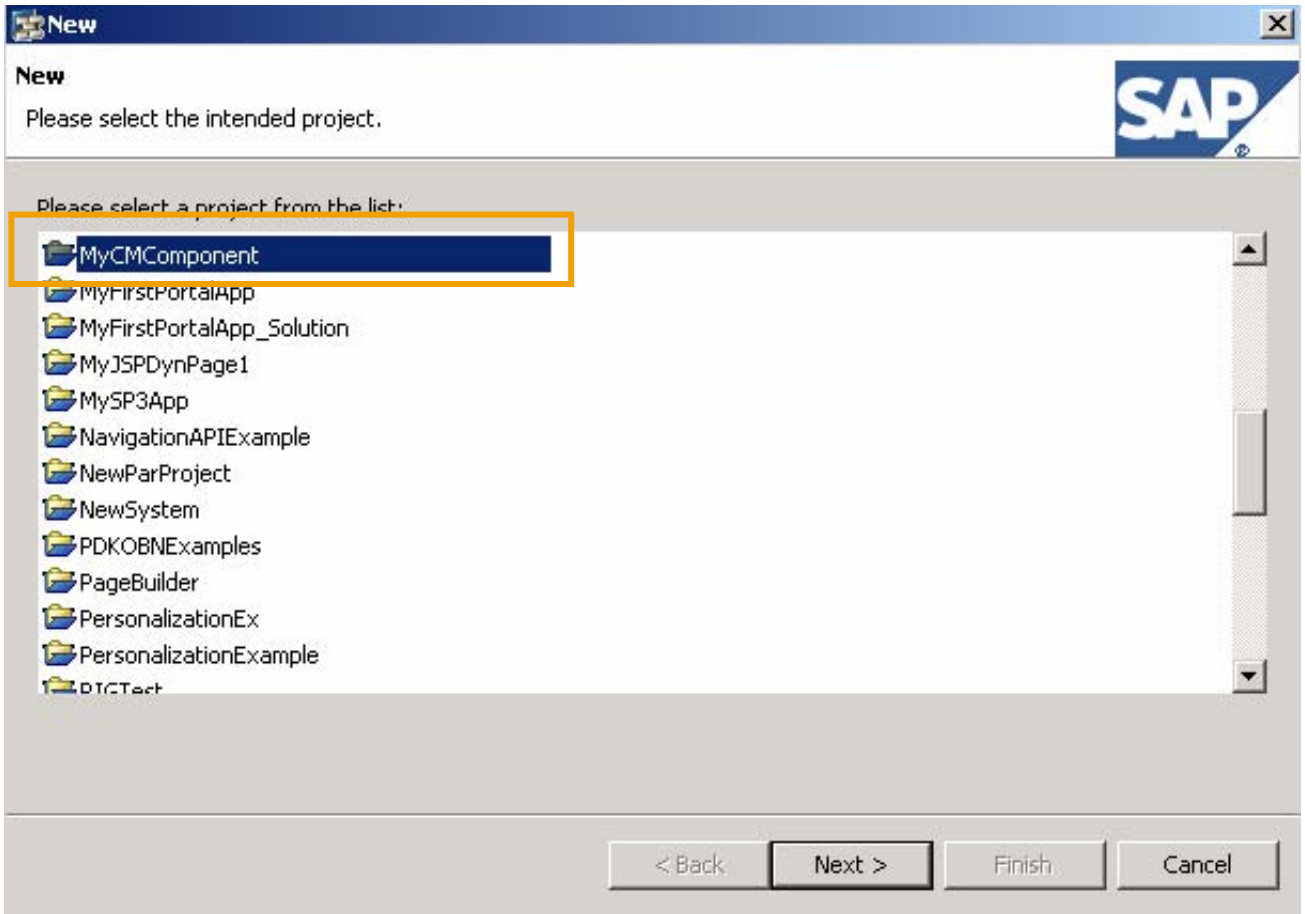
OR KM wizards directly via icons in the toolbar

OR right-clicking your project



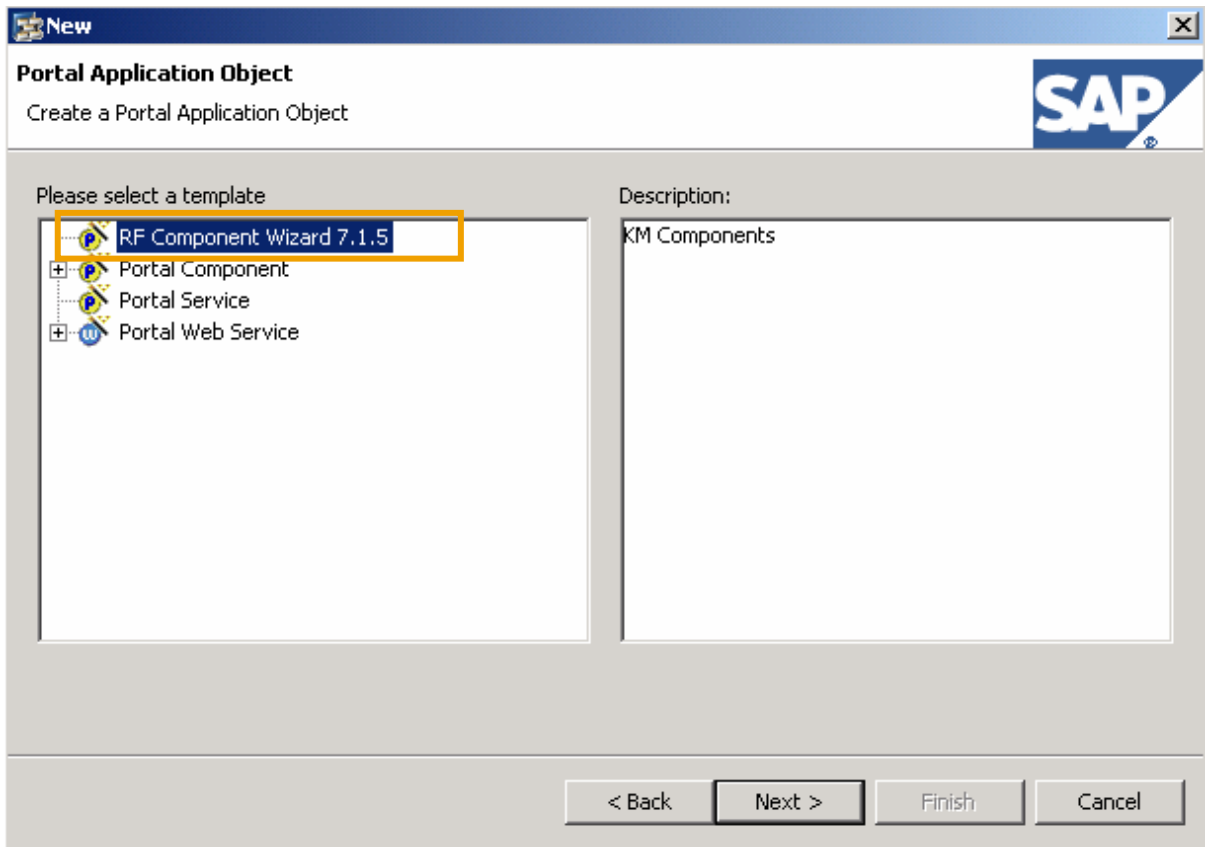
Create a new Portal Application Object (2)

Select the Portal Application Project and click „Next“:



Create a new Portal Application Object (3)

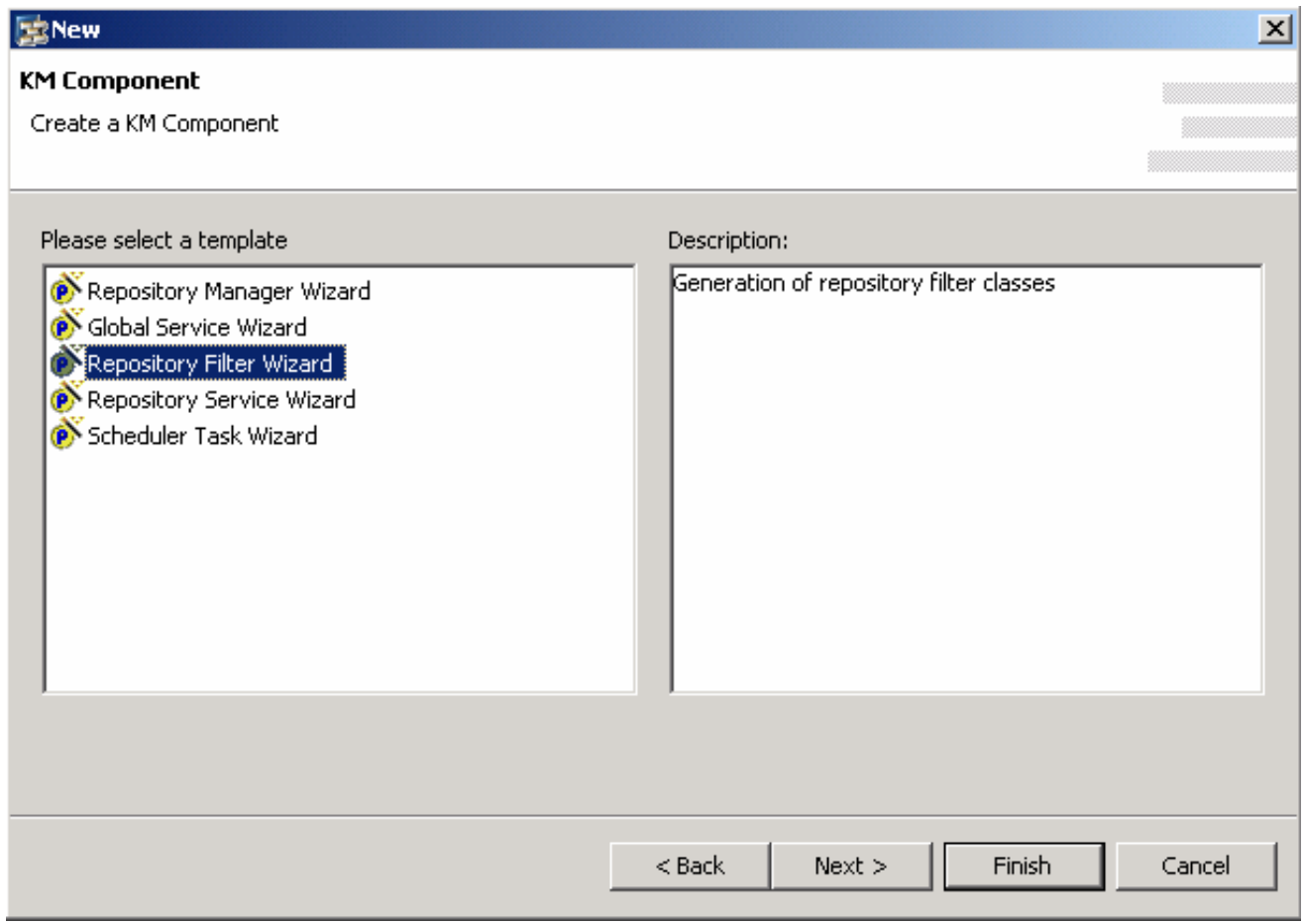
Create a new Repository Framework Component for your Portal Application Project:



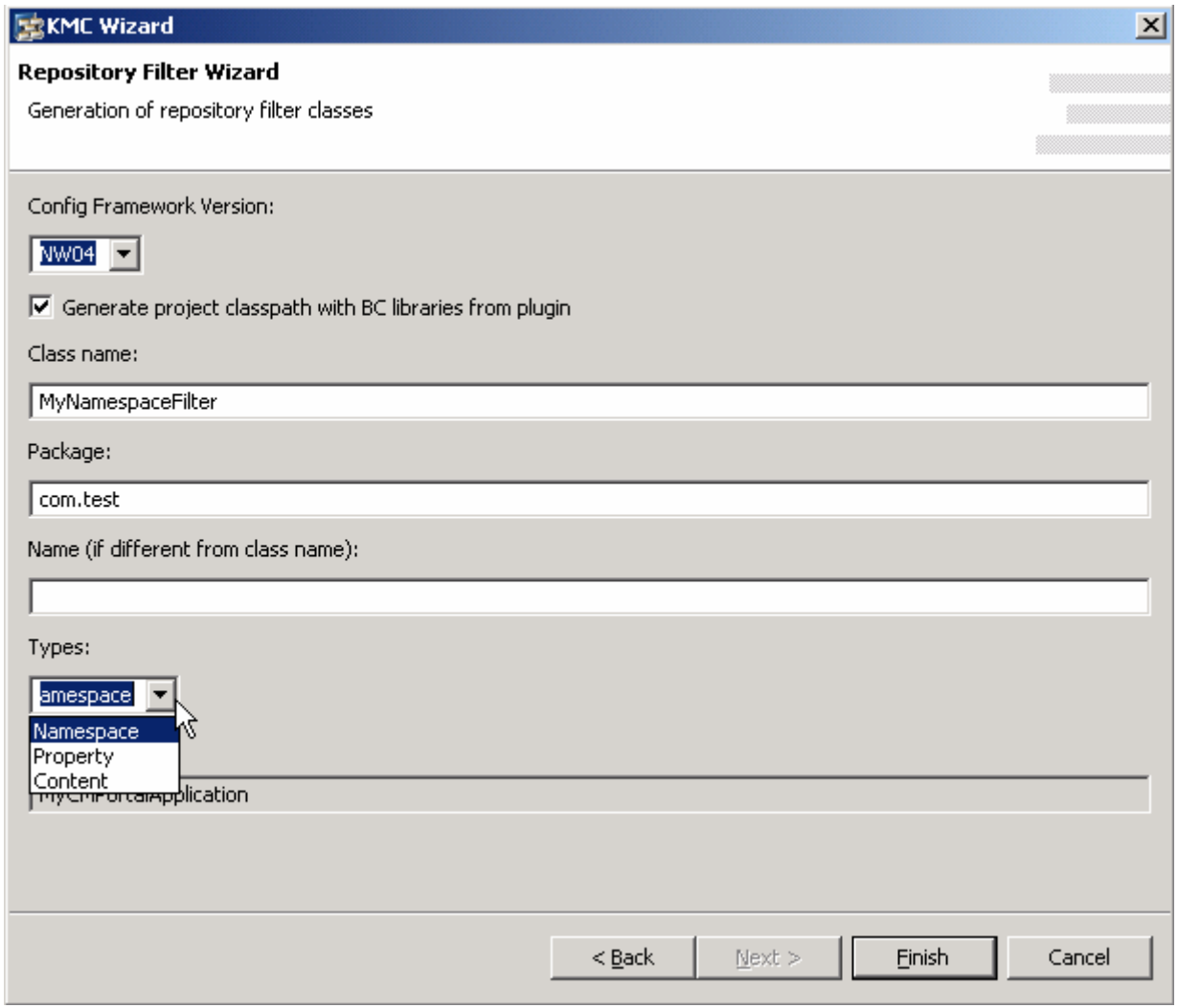
Create a new Portal Application Object (4)

Select which KM Component you want to implement:

- Repository Manager
- Global Service
- Repository Filter
- Repository Service
- Scheduler Task

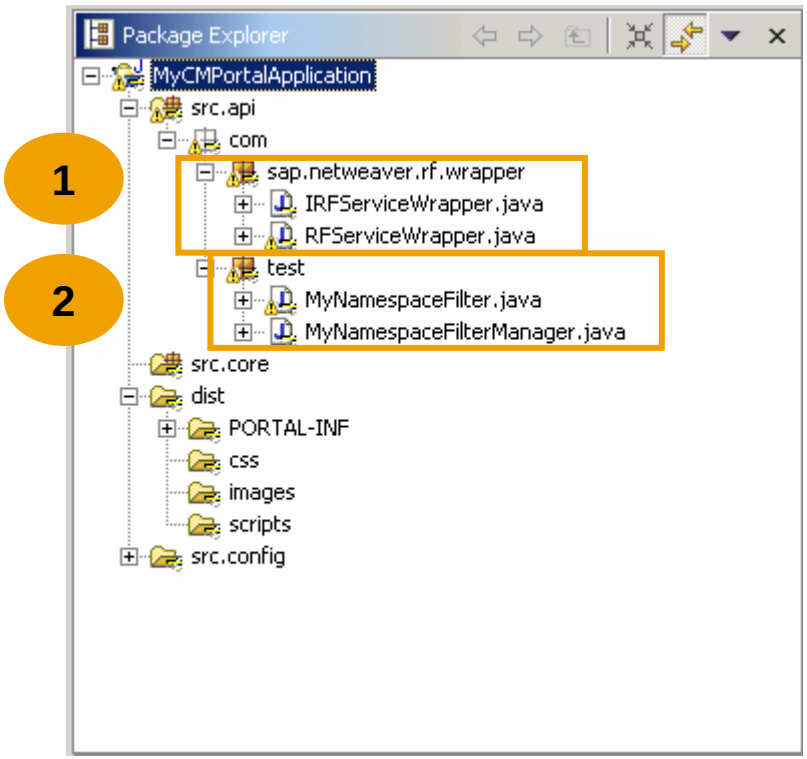


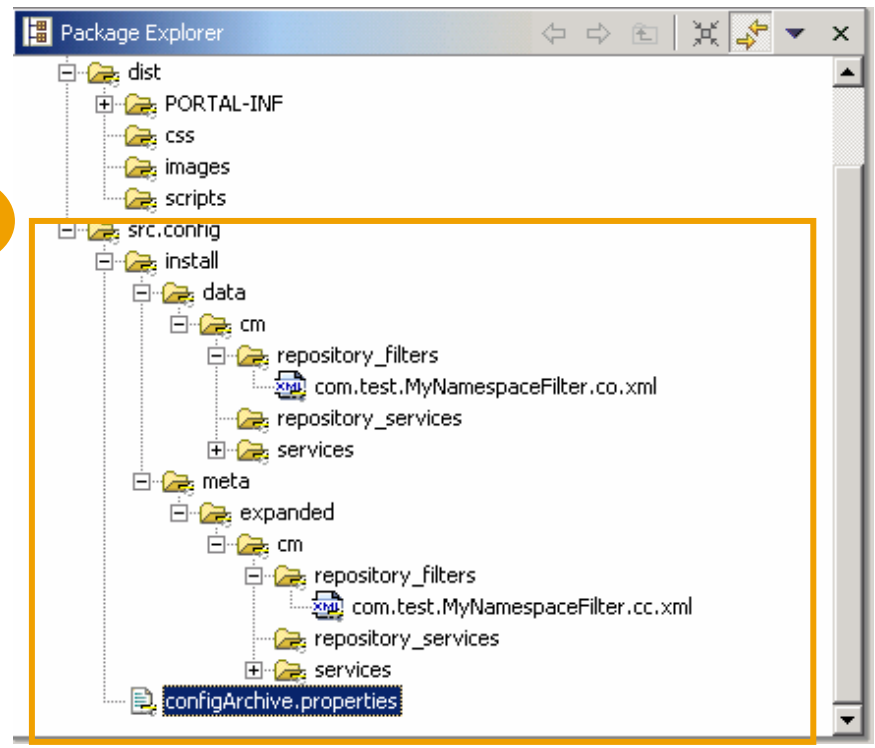
Example: Create a Repository Filter ➔ Namespace Filter



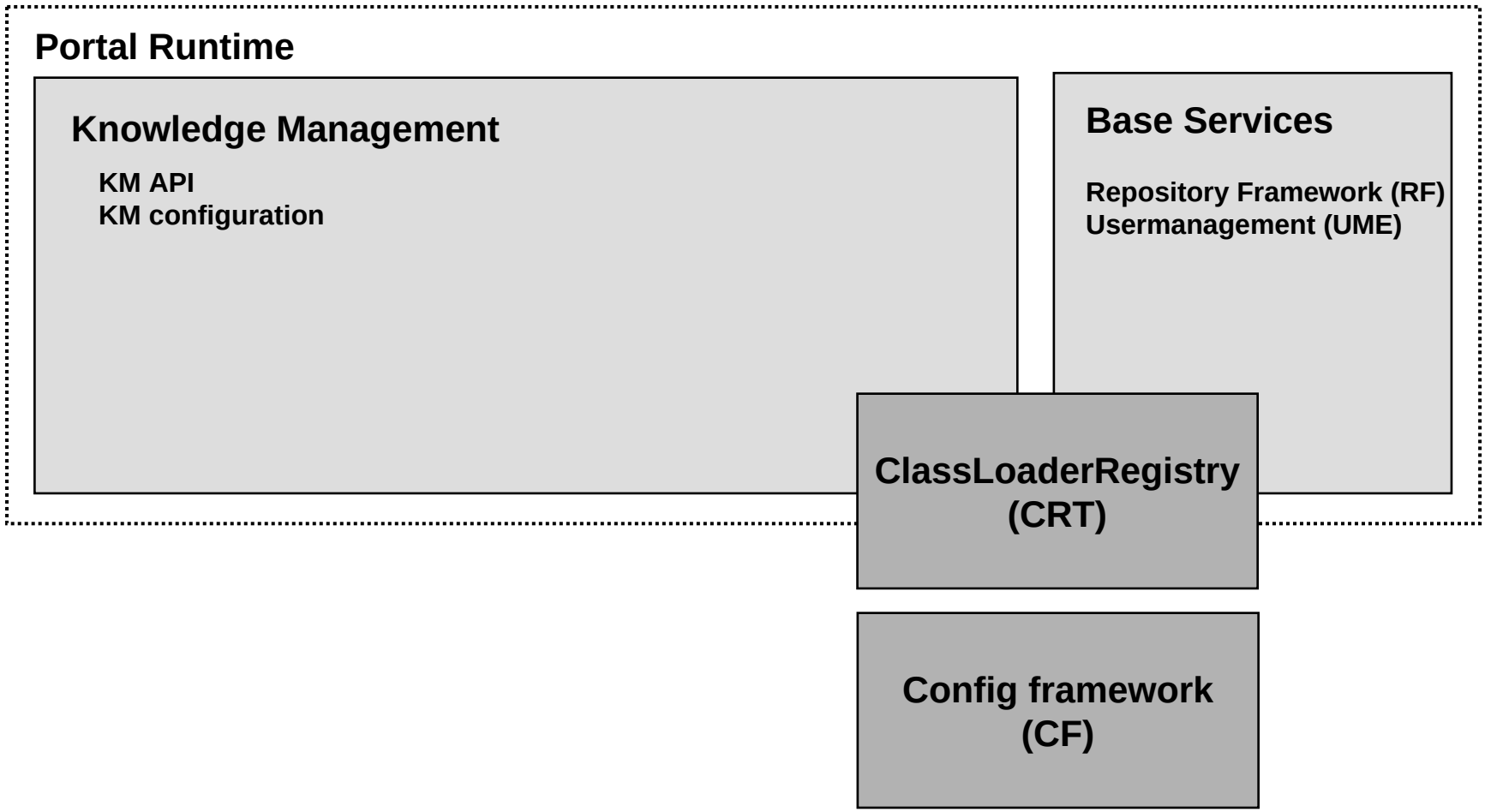
Project structure

1. **Portal service wrapper in src.API**
2. **CM Component classes**
3. **Config Archive for config framework**

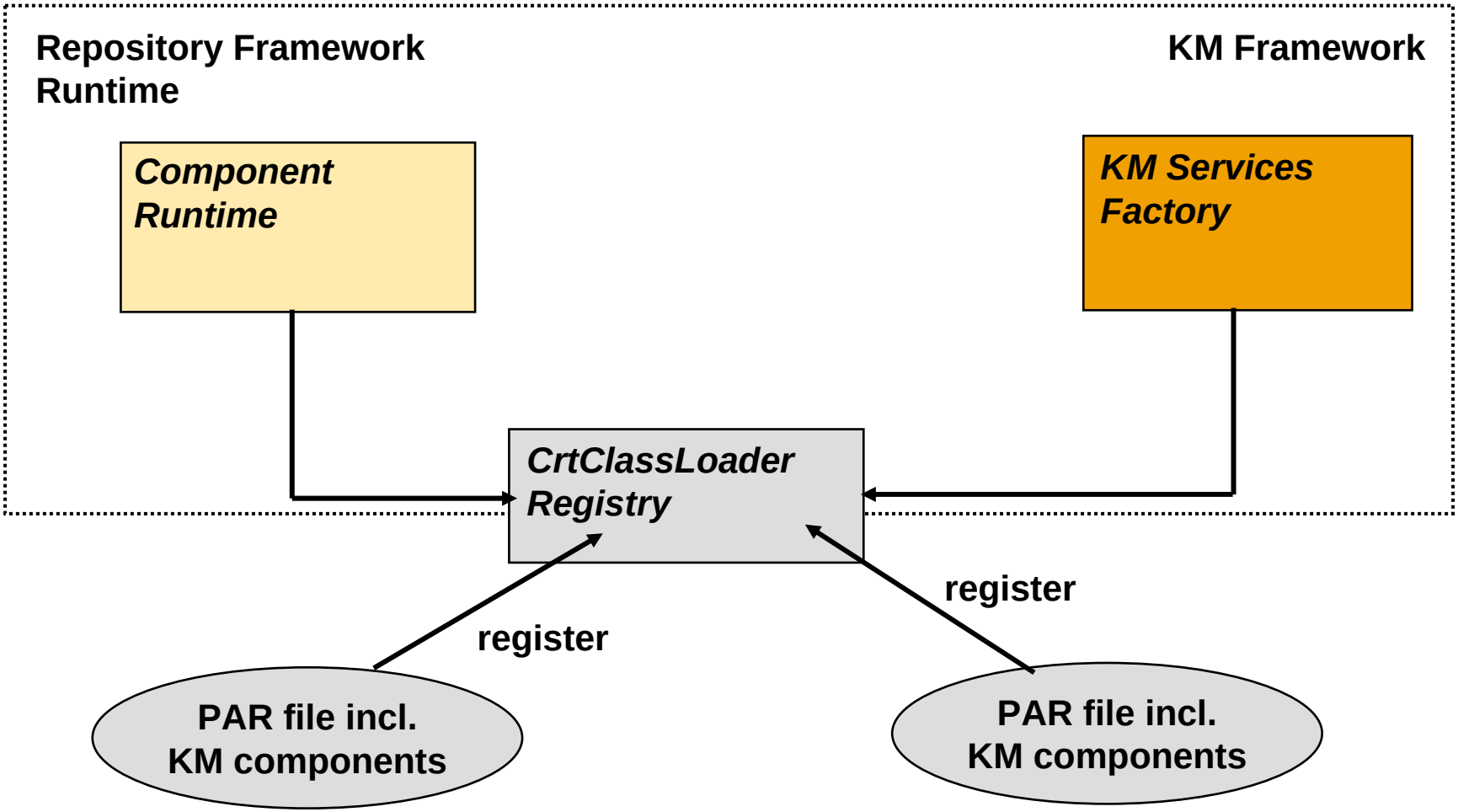




Neccessary components for KM component implementation & deployment process



CrtClassLoader Registration



A standard Portal Service with interface and implementation class uses the CrtClassLoader Registration

The Service KEY has to be defined:

```
package com.sap.netweaver.rf.wrapper;

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

public interface IRFServiceWrapper extends IService
{
    public static final String KEY = "/* To be inserted */";
}
```

Portal Service KEY:= <Servicename>

Naming convention:
<ServiceName> := <ApplicationName>.<Service>
<ApplicationName> := parfilename

A standard Portal Service implements the CrtClassLoader Registration

```
public class RFServiceWrapper implements
                              IRFServiceWrapper {

   public void init(IServiceContext serviceContext){
       mm_serviceContext = serviceContext;
       CrtClassLoaderRegistry.addClassLoader(
             this.getKey(), this.getClass().getClassLoader()
       );
   }
}
```

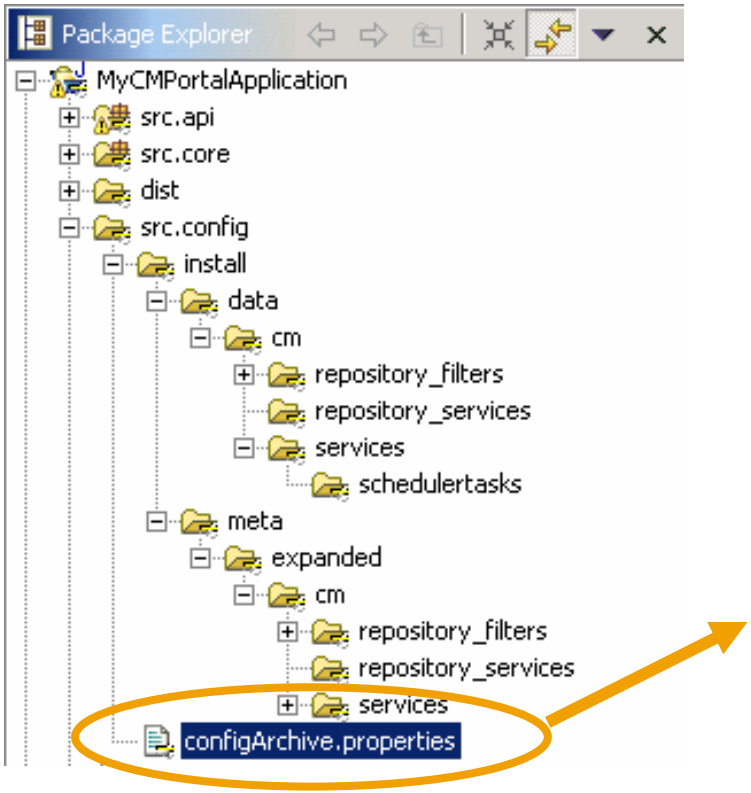
Registers this service wrappers classloader at the KM CrtClassLoader registry.

➔ **Mandatory implementation for all KM components**

Using the configuration framework (KM base services)

- KM components are exposed as configurables
- Access the configuration data with base service wrapper classes (IConfigurable)
- Deploy pre-defined Classes definitions and configurable instances with a KM component PAR files

Create / Check KM component configuration



Structure:

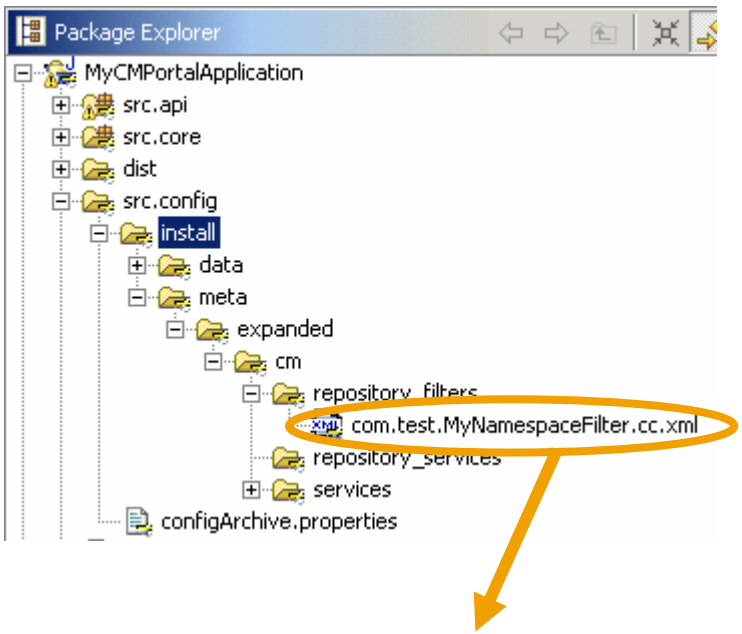
src.config/install/data/cm/…
src.config/install/meta/expanded/cm/
…

configArchive.properties:

ca.name=MyCMPortalApplication.prjconfig
ca.version=6.0.1.1
ca.creation.time=1117
ca.creation.date=20040826
ca.creation.user=unknown
ca.creation.machine=unknown
ca.dependencies: bc.util.prjconfig, bc.sf.prjconfig, bc.sf.service.prjconfig, bc.rf.prjconfig

cma.name=MyCMPortalApplication.prjconfig
cma.version=6.0.1.1
cma.storage=sfs
cma.creation.time=1117
cma.creation.date=20040826
cma.creation.user=unknown
cma.creation.machine=unknown
cma.dependencies: bc.util.prjconfig, bc.sf.prjconfig, bc.sf.service.prjconfig, bc.rf.prjconfig

Create / Check KM component configuration



1. Contains the class definition
2. Contains instance files

2. Instance definition (com.test.MyNamespaceFilter.co.xml):

```
<Configurable configclass="com.test.MyNamespaceFilter">
 <property name="name"
          value="com.test.MyNamespaceFilter" />
 <property name="active" value="true" />
 <property name="description" />
</Configurable>
```

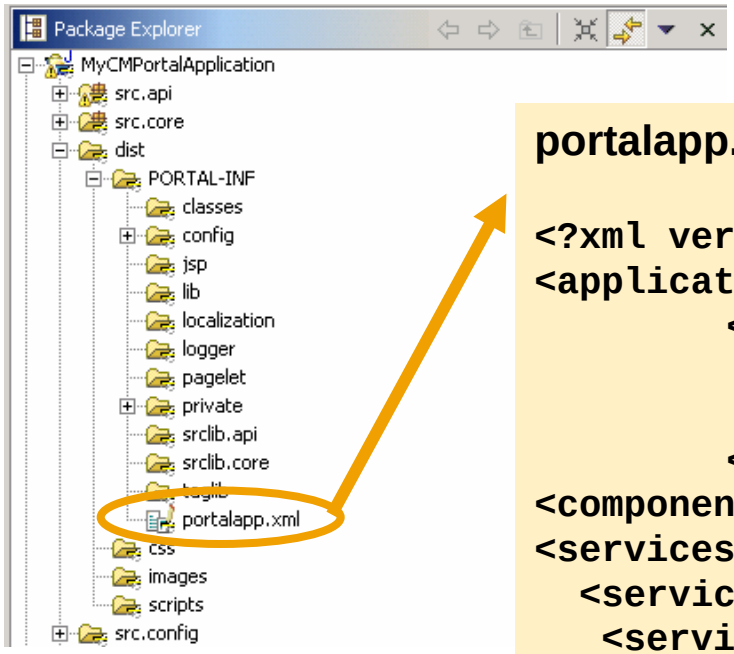
1. Class definition for this namespace filter (com.test.MyNamespaceFilter.cc.xml):

```
<ConfigClass name="com.test.MyNamespaceFilter"
         extends="RepositoryFilter">
 <attribute name="class" type="class"
         constant="com.test.MyNamespaceFilterManager"/>
</ConfigClass>
```

Deployment using PAR Export Wizards

- **Standard portal deployment**
  - **maintained portalapps.xml is mandatory**
  - **additional information can be provided in the public part of this PAR file**
- **Creates a new wrapper service for each PAR file including KM components**
  - **Service has no funcionality, except of classloader registration**
- **Configuration for configuration framework**
  - **automatically bundle the archive containíng**
    - **meta data (Classes definitions)**
    - **instances**

Create / Check portal service configuration

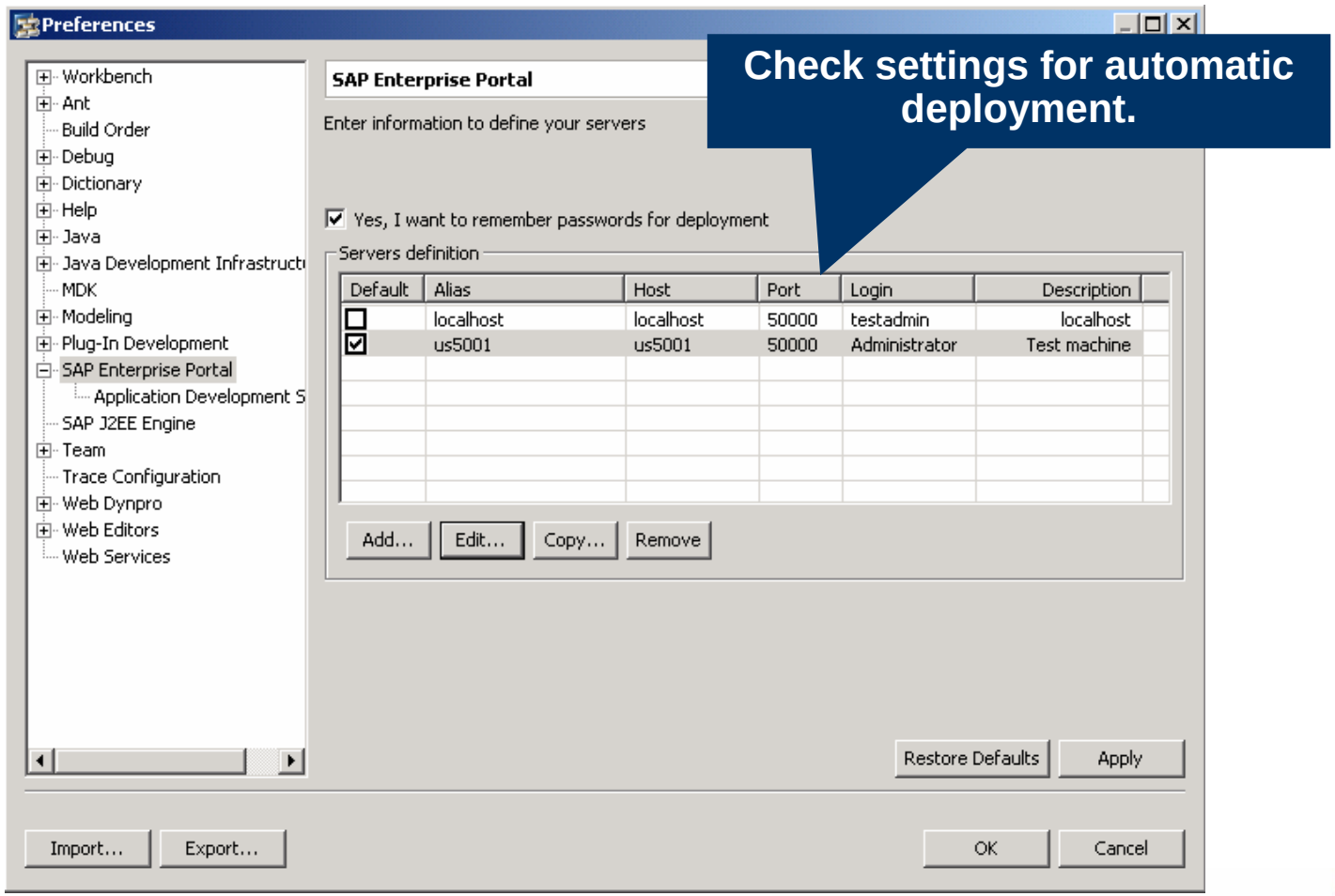


portalapp.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<application>
        <application-config>
                <property name="SharingReference"
                value="knowledgemanagement, …"/>
        </application-config>
<components/>
<services>
  <service name="RFServiceWrapper">
   <service-config>
    <property name="className"
value="com.sap.netweaver.rf.wrapper.RFServiceWrapper"/>
    <property name="startup" value="true"/>
   </service-config>
  </service></services>
</application>
```

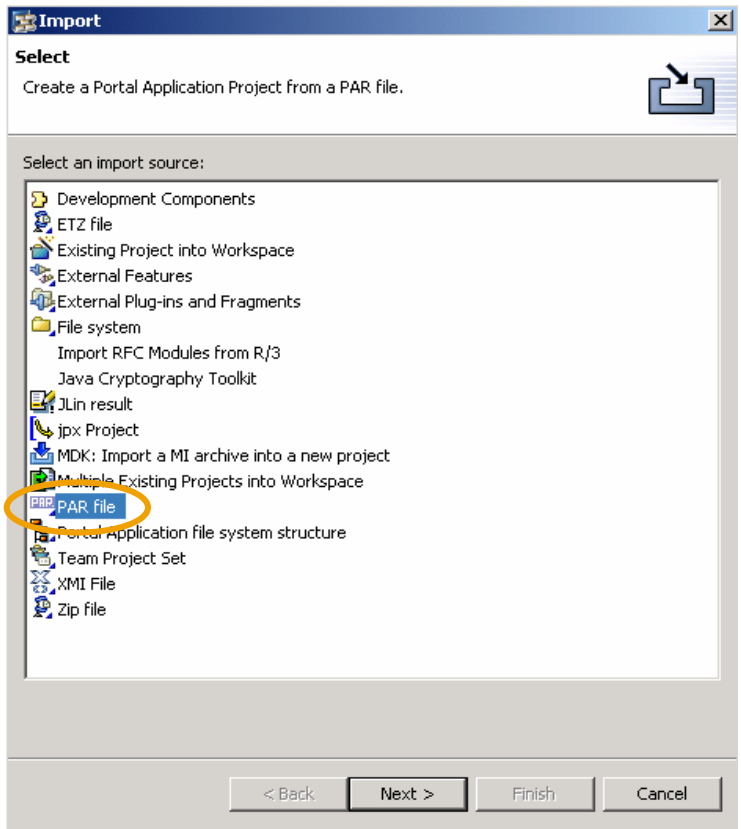
Server configuration – Connection to local/remote Portal

- **From the menu, choose:**
  Window -> Preferences -> SAP Enterprise Portal



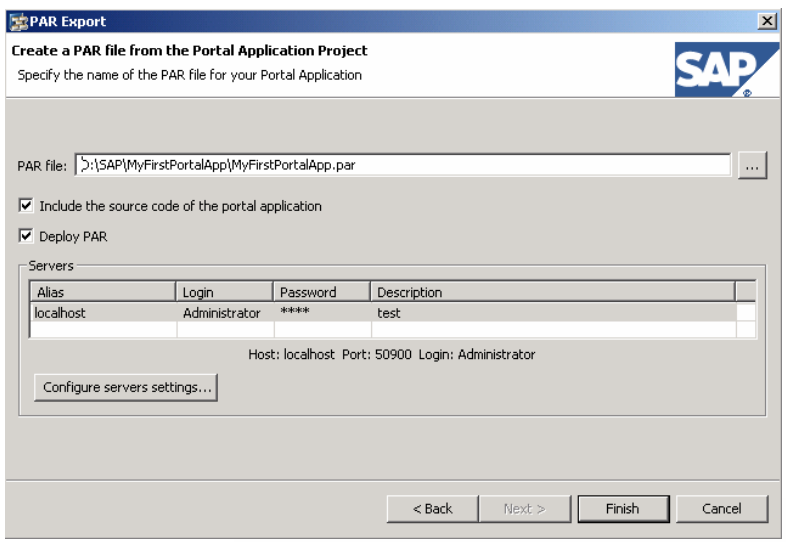
Create a PAR file

- From the menu, choose: File ➔ Export ➔ PAR File – click Next



Or choose the appropriate icon in the toolbar:

Or right-click your project and choose: 



After deployment ->restarting the SAP J2EE Engine is needed

Copyright 2005 SAP AG. All Rights Reserved