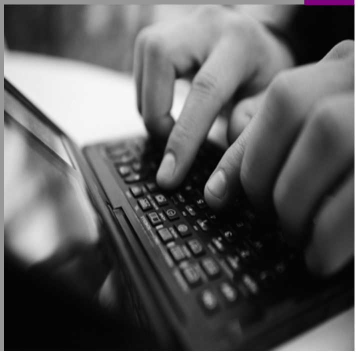
How-to Guide
SAP NetWeaver ‘04



How To…
Use the
BI Java SDK
in a Portal iView

Version 1.03 – July 2005

Applicable Releases:
SAP NetWeaver ’04, SP Stack 5 or greater
BI Java SDK, SP 12 or greater

THE BEST-RUN BUSINESSES RUN SAP

1 Scenario

You want to build a Java application that performs analytics on data in an XMLA provider and displays this data in an Enterprise Portal iView. You use the BI Java SDK to custom build this application and you establish the connection with the BI XMLA Connector. All you need for this scenario is included in SAP NetWeaver ’04.

For this scenario, be sure you have NetWeaver SP Stack 5 or greater installed. We also recommend that you have the Portal Development Kit (PDK) installed on your portal server.

2 Introduction

This document provides detailed instructions on how to use the BI Java SDK and its BI Java Connectors in an Enterprise Portal iView. It contains step-by-step instructions for creating an iView which uses the BI Java SDK’s BI XMLA Connector to connect to a BW system and retrieve a list of schemas.

In this document, we focus on the procedure for establishing a connection in the Enterprise Portal environment. After you establish this connection, refer to the tutorials and documentation for the BI Java SDK and the Portal Development Kit (PDK) to expand upon your scenarios.

3 The Step By Step Solution

In the first section below you prepare your system, and in the second section you create your iView.

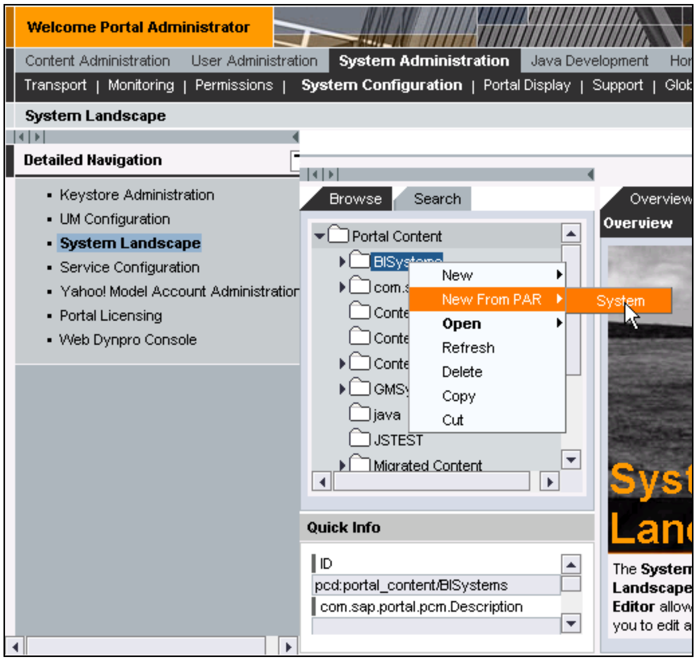
3.1 Prepare the System

This example uses the BI XMLA Connector to connect to an SAP BW system. Preparing your system consists of the following tasks:

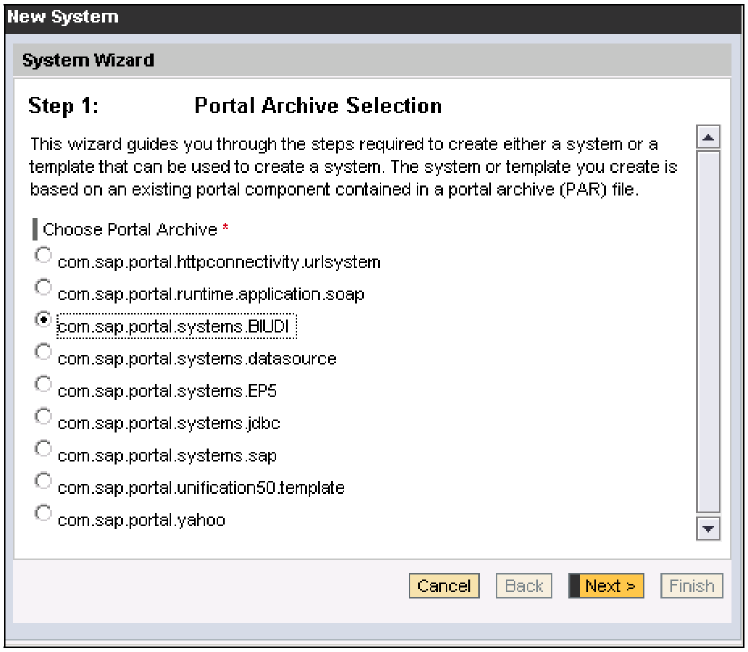
- Create a system in the Portal System Landscape for the XMLA provider to which you wish to connect
- Prepare the connection properties
- Configure the user mapping

To create a new system in the Portal System Landscape:

1. Log on to the Enterprise Portal and select *System Administration → System Configuration → System Landscape.*
2. On the *Browse* tab, select a system in the *Portal Content* folder, right-click, and choose *New From PAR → System* from the context menu.



3. In the System Wizard, on the *Portal Archive Selection* screen, select `com.sap.portal.systems.BIUDI`, and press *Next.*



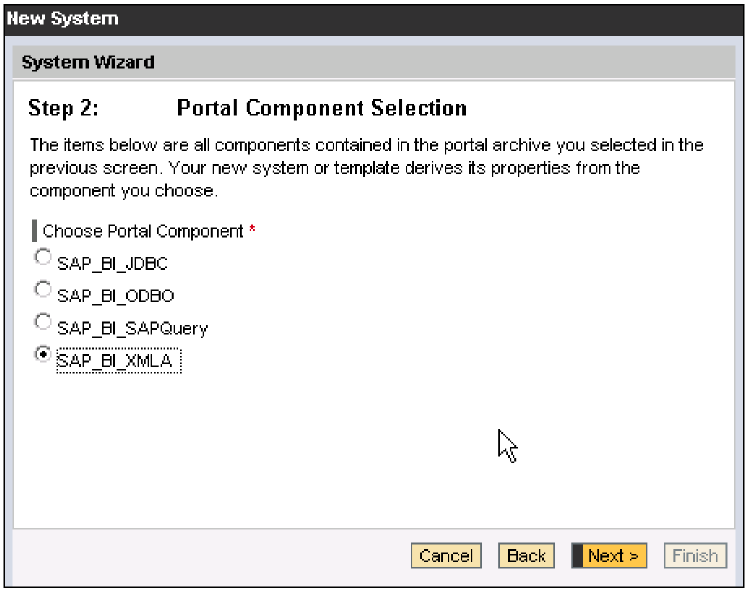
Step 1: Portal Archive Selection

This wizard guides you through the steps required to create either a system or a template that can be used to create a system. The system or template you create is based on an existing portal component contained in a portal archive (PAR) file.

Choose Portal Archive *

○ com.sap.portal.httpconnectivity.urlsystem
○ com.sap.portal.runtime.application.soap
● com.sap.portal.systems.BIUDI
○ com.sap.portal.systems.datasource
○ com.sap.portal.systems.EP5
○ com.sap.portal.systems.jdbc
○ com.sap.portal.systems.sap
○ com.sap.portal.unification50.template
○ com.sap.portal.yahoo

4. On the *Portal Component Selection* screen, select `SAP_BI_XMLA` (this corresponds to the BI XMLA Connector that we use to connect to the BW data source) and press *Next.*



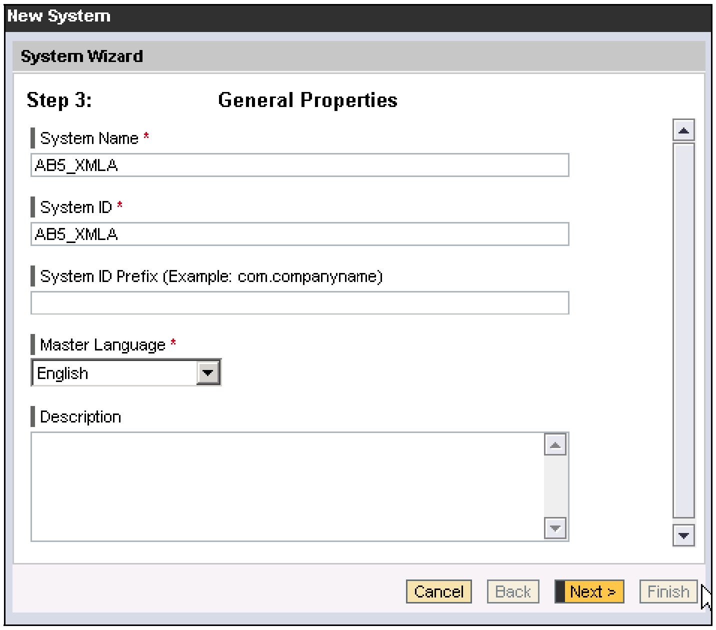
Step 2: Portal Component Selection

The items below are all components contained in the portal archive you selected in the previous screen. Your new system or template derives its properties from the component you choose.

Choose Portal Component *

○ SAP_BI_JDBC
○ SAP_BI_ODBO
○ SAP_BI_SAPQuery
● SAP_BI_XMLA

5. On the *General Properties* screen, configure the *System Name* and *System ID* fields with the values of your choice for your data source, and press *Next.*
6. Press *Finish* to create your system.



System Name * AB5_XMLA

System ID * AB5_XMLA

System ID Prefix (Example: com.companyname)

Master Language * English

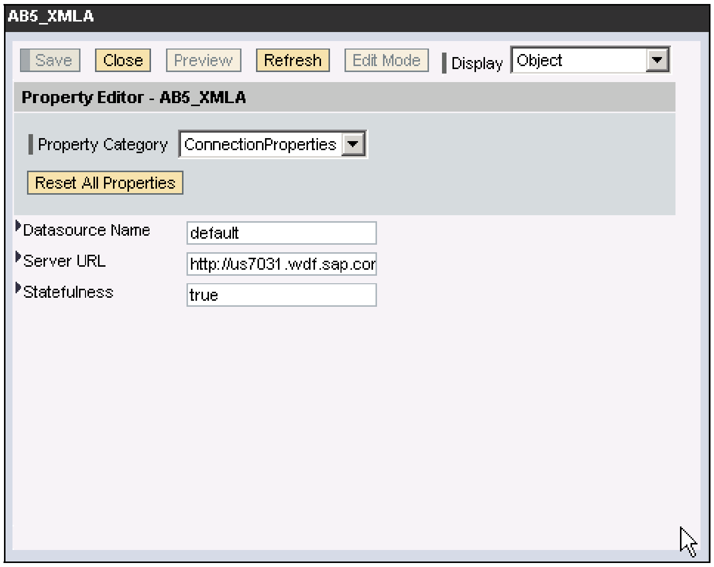
Description

To configure the connection properties:

7. On the *<system name>* tab, open the system for edit by choosing *Open the object for editing*, and press *OK.*
8. From the *Property Category* drop-down list, select *ConnectionProperties*, and configure the connection properties as pictured to the right. Enter your own server URL, which takes the format below:

   `http://server:port/sap/bw/xml/soap/xmla`

   Press *Save* to save the values.



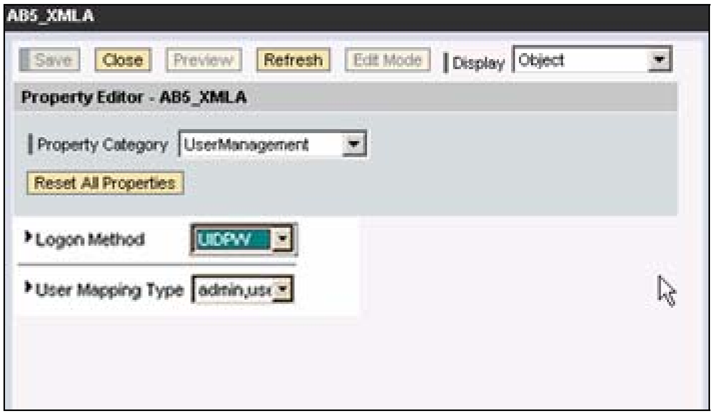
Display Object

Property Category ConnectionProperties

Datasource Name	default
Server URL	http://us7031.wdf.sap.cor
Statefulness	true

9. From the *Property Category* drop-down list, select *UserManagement*. From the *Logon Method* drop-down list, select *UIDPW*. From the *User Mapping Type* drop-down list, select *admin,user.*

   Press *Save* to save the value.



Property Editor - AB5_XMLA

Display: Object

Property Category: UserManagement

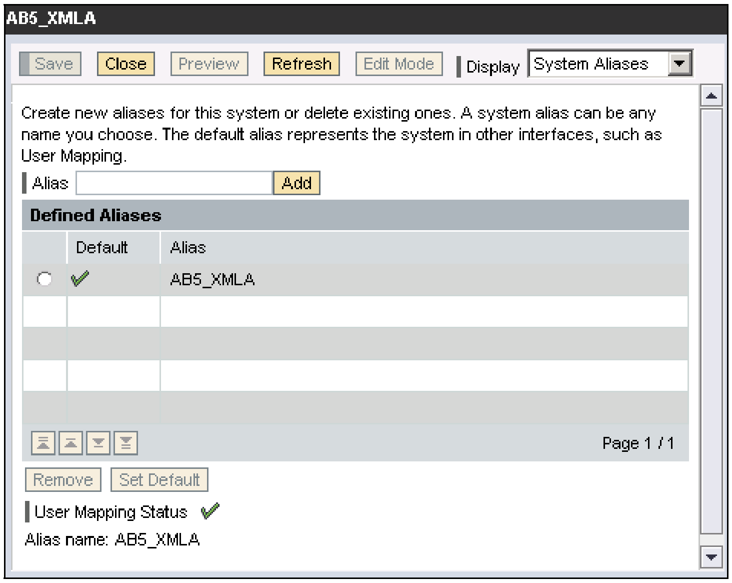
Logon Method: UIDPW

User Mapping Type: admin,us

10. Create a new alias for this system, which will be used to get the connection in the Java source code for your application:

    Using the *Display* drop-down list, select *System Aliases* and add an alias to your system. In this example, we name the alias *AB5_XMLA*.

    Press *Save* to save the value.



Display: System Aliases

Create new aliases for this system or delete existing ones. A system alias can be any name you choose. The default alias represents the system in other interfaces, such as User Mapping.

Alias:

Defined Aliases

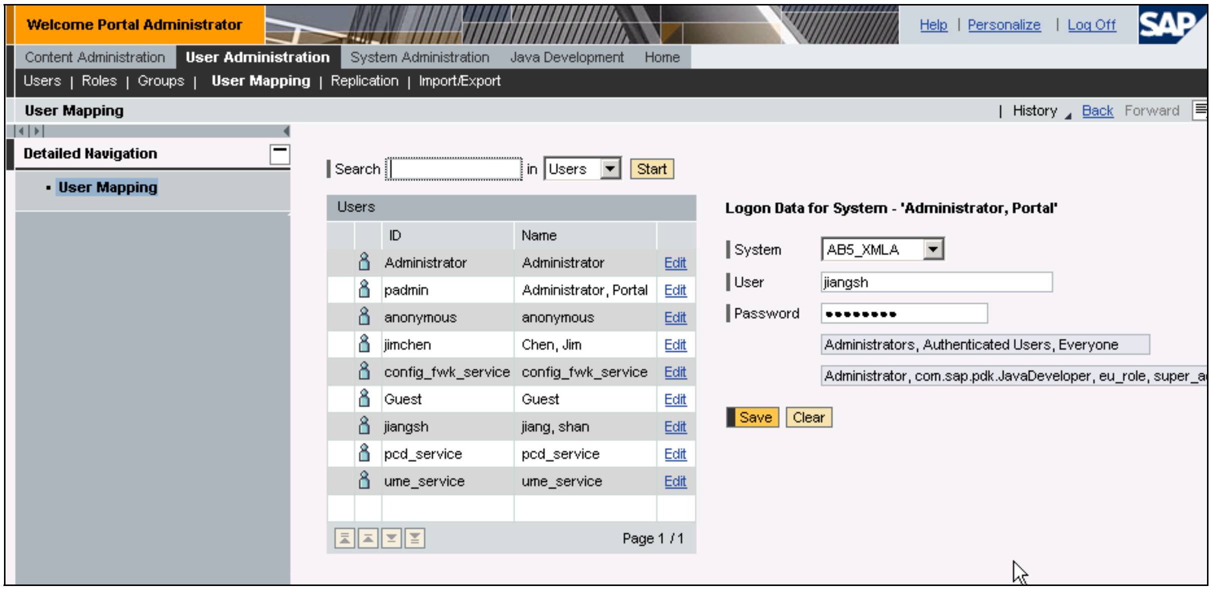
	Default	Alias
○	✓	AB5_XMLA

User Mapping Status ✓

Alias name: AB5_XMLA

To configure user mapping:

11. From the first-level Portal navigation, select *User Administration* → *User Mapping*, and map your Portal user to the user of the newly created system (in this case, the user of the target BW system), also configuring your user name and password for your target system.



Users

ID	Name	
Administrator	Administrator	Edit
padmin	Administrator, Portal	Edit
anonymous	anonymous	Edit
jimchen	Chen, Jim	Edit
config_fwk_service	config_fwk_service	Edit
Guest	Guest	Edit
jiangsh	jiang, shan	Edit
pcd_service	pcd_service	Edit
ume_service	ume_service	Edit

Page 1 / 1

Logon Data for System - 'Administrator, Portal'

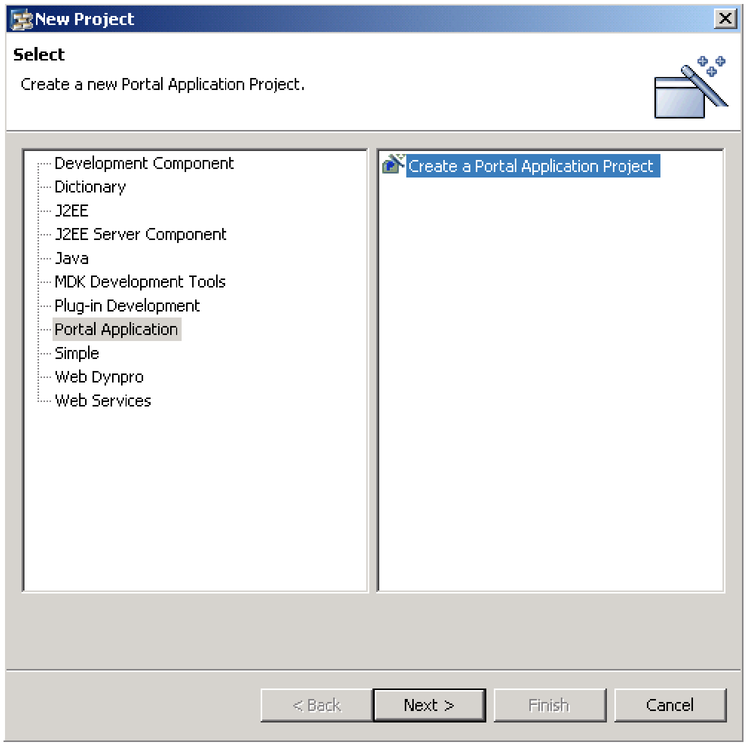
System AB5_XMLA
User jiangsh
Password ••••••••
Administrators, Authenticated Users, Everyone
Administrator, com.sap.pdk.JavaDeveloper, eu_role, super_a

Now your system is prepared and ready for you to create your iView.

3.2 Create the Sample iView

Create a new iView in the SAP NetWeaver Developer Studio:

12. In the *New Project Wizard*, select *Portal Application*, choose *Create a Portal Application Project*, and follow the steps to create a project.



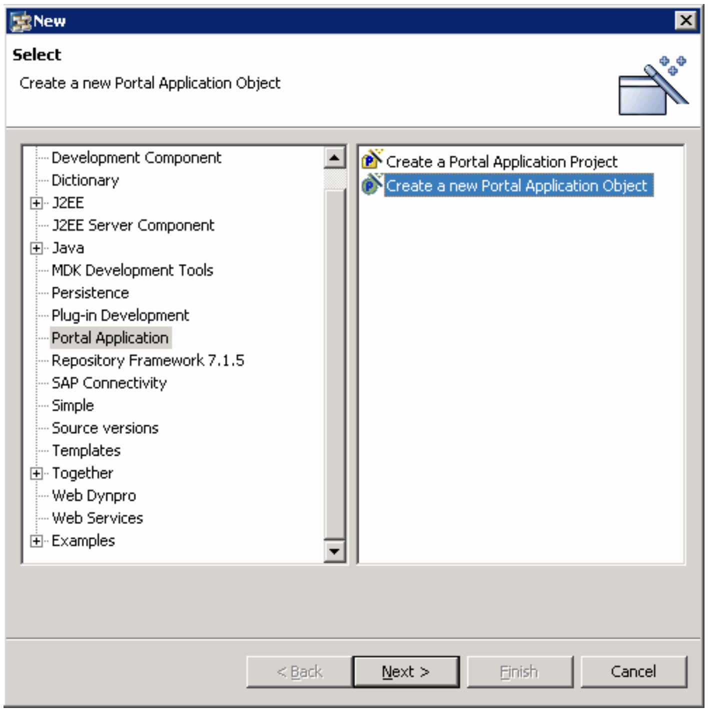
New Project

Select

Create a new Portal Application Project.

13. In the *Package Explorer*, right-click on the new Portal project, and select *New* → *Other* from the context menu.

14. In the wizard, select *Portal Application*, choose *Create a new Portal Application Object*, and click *Next*.



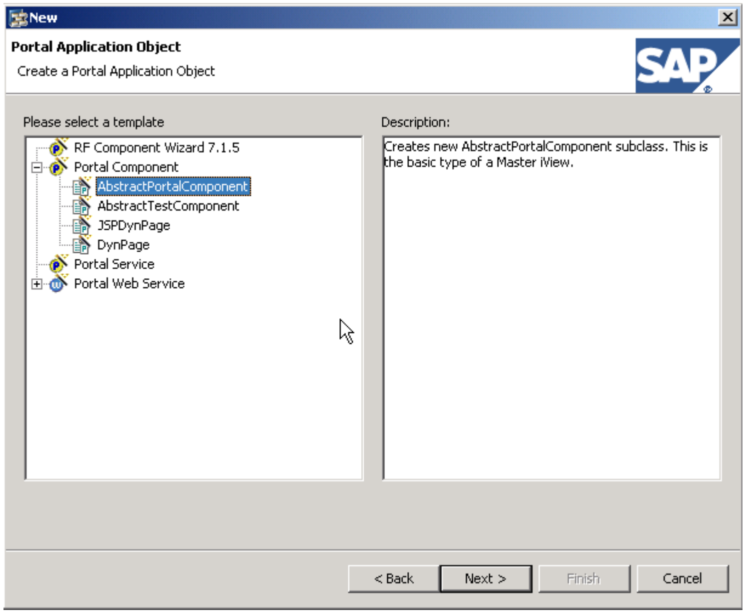
New

Select

Create a new Portal Application Object

15. Select your Portal project from the list, and click *Next*.

16. Expand the *Portal Component* node and choose *AbstractPortalComponent* as the template. Click *Next.*

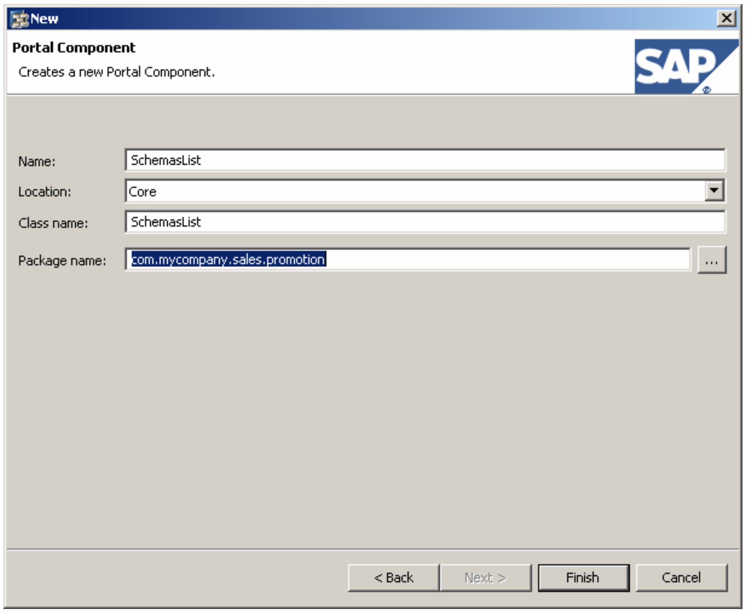


Portal Application Object

Create a Portal Application Object

Description: Creates new AbstractPortalComponent subclass. This is the basic type of a Master iView.

17. Configure values for your Portal component on the *Portal Component* screen, and click *Finish* to create your Portal component.



Portal Component

Creates a new Portal Component.

Name: SchemasList
Location: Core
Class name: SchemasList
Package name: com.mycompany.sales.promotion

18. Locate and add required libraries to your project:

    - BI Java SDK libraries:

      If you don't already have the BI Java SDK, go to:
      `<your NetWeaver Developer Studio installation folder>\eclipse\tools\bi_sdk\bi_sdk.zip`,

      or download it from the SAP Developer Network at http://www.sdn.sap.com/.

    - Portal Connector Gateway Service library:

      `com.sap.portal.ivs.connectorserviceapi.jar` or
      `com.sap.portal.ivs.connectorservice_api.jar` (SP Stack 09 or greater)

      If you have the Portal Development Kit installed on your portal server, you can find this library in *Java Development* → *Downloads* → *Portal Libraries* → libraries.zip.

      Otherwise, look in:

      `http://<server>:<port>/irj/servlet/prt/portal/prtroot/com.sap.portal.support.browse.default`,

      and then follow this path:
      `ROOT/WEB-INF/portal/portalapps/com.sap.portal.ivs.connectorservice/lib`

      Click the *download* link to download a zip file which contains this jar.

19. In the iView source code, use the Portal Connector Gateway Service to get a connection, specifying the system alias you created above in Section 3.1, Step 10 (`AB5_XMLA` in our example). Use the code below as an example to get the connection and retrieve schemas:

```
IConnectorService connectorService;
IConnectorGatewayService connectorGatewayService;
IConnection connection;
try {
  connectorService =
    (IConnectorService) PortalRuntime
      .getRuntimeResources()
      .getService(
      IConnectorService.KEY);
  connectorGatewayService =
    connectorService.getIConnectorGatewayService();
```

```
      ConnectionProperties connectionProperties =
        new ConnectionProperties(
          request.getLocale(),
          request.getUser());
      connection =
        connectorGatewayService.getConnection(
          "AB5_XMLA",
          connectionProperties);

      IBIOlap olap = ((IBIConnection) connection).getOlap();

      List schemas = olap.getSchema();
      Schema schema = null;

      for (int i = 0; i < schemas.size(); i++) {
        schema = (Schema) schemas.get(i);
        response.getWriter().write(
          "schema: " + schema.getName() + "<br>");
      }
      connection.close();

    } catch (Exception e) {
      e.printStackTrace();
    }
```

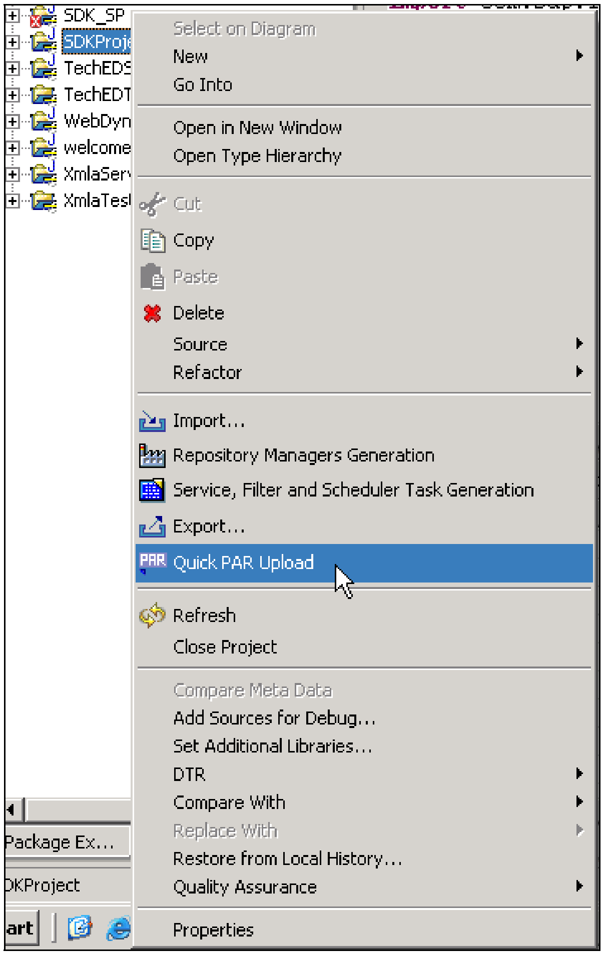
20. Edit the portalapp.xml file of this iView to add required references:

```
  <application-config>
    <property name="SharingReference"
value="SAPJ2EE::library:com.sap.ip.bi.sdk,SAPJ2EE::library:bi~mmr~jmi,SAPJ2
EE::library:bi~mmr~core,SAPJ2EE::library:bi~mmr~db,SAPJ2EE::library:bi~mmr~
cwm_1.0_library,com.sap.portal.htmlb,com.sap.portal.ivs.connectorservice"/>
  </application-config>
```

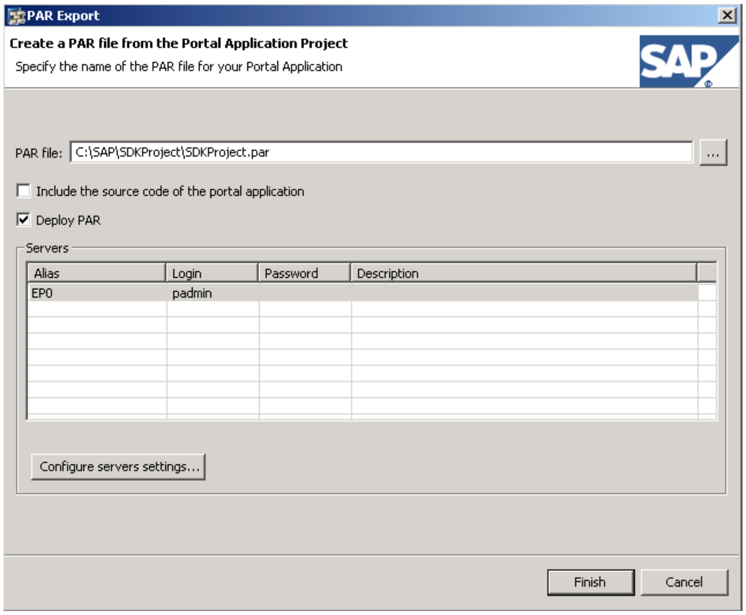
Note: You may need to configure your portalapp.xml file with additional properties, such as `SecurityZone`, as required by your system. See the Portal Development Kit documentation for further details, and a full sample portalapp.xml file in the Appendix of this document.

Build the PAR file and upload it into the Portal:

21. In the *Package Explorer*, from the context menu of your Portal project, select *Quick PAR Upload.*



22. On the *PAR Export* screen, select the *Deploy PAR* checkbox, configure your server settings as needed, and click *Finish*.



Create a PAR file from the Portal Application Project

Specify the name of the PAR file for your Portal Application

PAR file: C:\SAP\SDKProject\SDKProject.par

☐ Include the source code of the portal application

☑ Deploy PAR

Servers

Alias	Login	Password	Description
EP0	padmin		

Now, you can run your iView, which retrieves a list of schemas in your system. For more information about building and deploying iViews, see the Portal Development Kit documentation.

4 Appendix: Source Code

Below is source code for the Portal component class and portalclass.xml files that demonstrate this scenario.

4.1 Portal component class

```
package com.mycompany.sales.promotion;

import java.util.List;

import org.omg.cwm.analysis.olap.Schema;

import com.sap.ip.bi.sdk.dac.connector.IBIConnection;
import com.sap.ip.bi.sdk.dac.connector.IBIOlap;
import com.sapportals.connector.connection.IConnection;
import com.sapportals.portal.ivs.cg.ConnectionProperties;
import com.sapportals.portal.ivs.cg.IConnectorGatewayService;
import com.sapportals.portal.ivs.cg.IConnectorService;
import com.sapportals.portal.prt.component.AbstractPortalComponent;
import com.sapportals.portal.prt.component.IPortalComponentRequest;
import com.sapportals.portal.prt.component.IPortalComponentResponse;
import com.sapportals.portal.prt.runtime.PortalRuntime;
public class SchemasList extends AbstractPortalComponent {
  public void doContent(
    IPortalComponentRequest request,
    IPortalComponentResponse response) {
    IConnectorService connectorService;
    IConnectorGatewayService connectorGatewayService;
    IConnection connection = null;
    try {
      connectorService =
        (IConnectorService) PortalRuntime
          .getRuntimeResources()
          .getService(
          IConnectorService.KEY);
      connectorGatewayService =
        connectorService.getIConnectorGatewayService();
      ConnectionProperties connectionProperties =
        new ConnectionProperties(
          request.getLocale(),
          request.getUser());
      connection =
        connectorGatewayService.getConnection(
          "AB5_XMLA",
          connectionProperties);

      IBIOlap olap = ((IBIConnection) connection).getOlap();

      List schemas = olap.getSchema();
      Schema schema = null;
```

```
   for (int i = 0; i < schemas.size(); i++) {
    schema = (Schema) schemas.get(i);
    response.getWriter().write(
     "schema: " + schema.getName() + "<br>");
   }

  } catch (Exception e) {
   e.printStackTrace();
  } finally {
   try {
    connection.close();
   } catch (Exception e) {
    e.printStackTrace();
   }
  }

 }
}
```

4.2 portalapp.xml file

Note: This example is based on Enterprise Portal version SP Stack 07. Check the Portal Development Kit documentation for more information on these settings.

```
<?xml version="1.0" encoding="utf-8"?>
<application>
 <application-config>
  <property name="SharingReference"
value="SAPJ2EE::library:com.sap.ip.bi.sdk,SAPJ2EE::library:bi~mmr~jmi,SAPJ
2EE::library:bi~mmr~core,SAPJ2EE::library:bi~mmr~db,SAPJ2EE::library:bi~mm
r~cwm_1.0_library,com.sap.portal.htmlb,com.sap.portal.ivs.connectorservice
"/>
 </application-config>
 <components>
  <component name="SchemasList">
   <component-config>
    <property name="ClassName"
value="com.mycompany.sales.promotion.SchemasList"/>
    <property name="SecurityZone" value="com.sap.portal/low_safety"/>
   </component-config>
   <component-profile/>
  </component>
 </components>
 <services/>
</application>
```

www.sap.com/netweaver