WebSphere. software

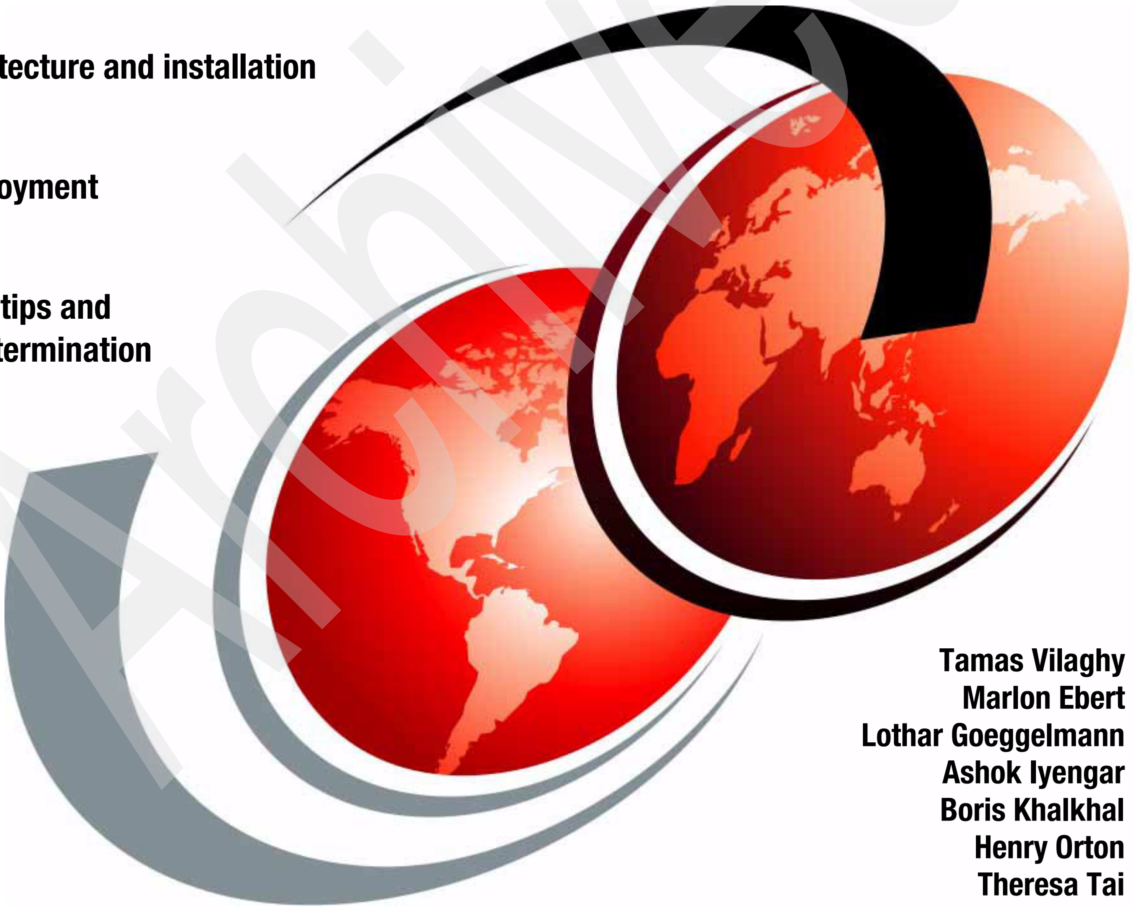
IBM

WebSphere Portal on z/OS

Portal architecture and installation

Portlet deployment

Installation tips and problem determination



Tamas Vilaghy
Marlon Ebert
Lothar Goeggelmann
Ashok Iyengar
Boris Khalkhal
Henry Orton
Theresa Tai

Redbooks

ibm.com/redbooks

International Technical Support Organization

WebSphere Portal on z/OS

November 2003

SG24-6992-00

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

First Edition (November 2003)

This edition applies to Version 4.1 of WebSphere Portal for z/OS.

Contents

Notices

Trademarks

Preface

This IBM® Redbook will help you install, tailor, and configure the PTF2 level of the WebSphere® Portal Server for z/OS® product. We discuss architectural, installation, configuration, administration, security, and problem determination issues. We show you how to deploy portlets and how to convert a “normal” WebSphere application into a portlet.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

The residency team (from left): Boris, Theresa, Tamas, Henry and Ashok

Marlon and Lothar from the Boeblingen Lab

Tamas Vilaghy is a Project Leader at the International Technical Support Organization, Poughkeepsie Center. He leads redbook projects dealing with e-business on IBM @server zSeries® servers. Before joining the ITSO, he worked in the System Sales Unit and Global Services departments of IBM Hungary. Tamas spent two years in Poughkeepsie from 1998 to 2000 involved with zSeries marketing and competitive analysis. From 1991 to 1998 he held technical, marketing and sales positions concerning zSeries. From 1979 to 1991 he was occupied with system software installation, development, and education.

Marlon Ebert is a Systems Engineer for z/OS and OS/390® at the IBM Development Lab in Boeblingen. He has 16 years of experience in the computing field. He started with mainframe operations and then went into S/36 and AS/400® programming. He spent three years supporting CRM software in the field and then joined z/OS and OS/390 Systems Engineering four years ago. His current areas of expertise include TCP/IP, UNIX System Services, LDAP, WebSphere Application Server and WebSphere Portal Server in the Parallel Sysplex® environment.

Lothar Goeggelmann is a Software Development Engineer in the IBM Development Lab in Boeblingen. He was the development lead for the project WebSphere Portal V4 on z/OS. Before this, he led the development team to port WebSphere Portal V4 to z/Linux. He has four years of experience in the development of the WebSphere Application Server on z/OS and in total 19 years of experience in software development in the Boeblingen Lab. He has a degree in

Computer Science from the Fachhochschule Ulm and is a certified Project Management Professional (PMP).

Ashok Iyengar is a Consultant with the WebSphere Enablement Team in San Diego, USA. He has 19 plus years of experience in the IT field. He holds a master's degree in Computer Science from North Dakota State University, Fargo. He has written extensively on WebSphere. He recently co-authored a book entitled *WebSphere Portal Primer* published by IBM Press.

Boris Khalkhal is an Advisory IT Specialist in La Gaude, France, in charge of pan-EMEA WebSphere for z/OS Technical Sales. He holds a bachelor's degree in Computer Science from the University of Nice, France. He has seven years of experience in the IT field including more than four years on WebSphere for z/OS. He has worked at IBM for four years.

Henry Orton is a Technical Sales I/T Consultant in the United Kingdom. He has many years experience in the areas of networking and communications and holds a degree in Pure Mathematics. His areas of expertise include software engineering, networking, telephony, multimedia and digital broadcasting. He has published papers on telecommunications and is a speaker for the IBM Software Group at technical conferences.

Theresa Tai works in zSeries WebSphere Marketing Technical Support in Poughkeepsie, New York. She has four years of experience in supporting WebSphere and Java on z/OS and OS/390. She holds a degree in Computer Science from Bernard Baruch College in New York City. Her background was in JES3 and JES2 design, development and service. She co-authored the redbook *WebSphere for z/OS V4 Problem Determination*.

Thanks to the following people for their contributions to this project:

Chen Chang, Office of the Attorney General, Texas, Austin
Mr. Chang participated in the residency redbook project for three weeks as a customer of WebSphere Portal Server for z/OS. He contributed valuable comments to the installation section, provided the "Installation worksheet" in Chapter 2 of this redbook, and also wrote the REXX script to convert RACF® userids to LDAP entries.

Rich Conway, William Tworek
International Technical Support Organization

Klaus Rindtorff, John T. Gates, Lothar Goeggelmann, Irene Stahl
IBM WebSphere Portal Development

Andre Watanabe
IBM Sweden

Doug Phillips
IBM Developer Technical Support Center, Dallas

Tom Hulsey, Dipak Shah, Ruben Salinas
IBM WebSphere Portal L2 support

Stanley Tzueing Shieh
IBM WebSphere World Wide Technical Sales Support

Ronald Lotter
IBM WebSphere Enablement Team

Dawn Hamilton, Frank DeGilio, Lynn Winkelbauer
IBM Design Center for e-business on-demand

Theresa Smit
WebSphere Portal content publishing Customer Partnership Team

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

WebSphere Portal architectures

In this chapter we introduce WebSphere Portal Enable V4.1 on the z/OS and OS/390 platforms. We give a general discussion of the logical tiers and architectures that is applicable to a portal running on zSeries. We wrap up our discussion with a description of physical architectures and also the architectures we used during this project.

1.1 WebSphere Portal

IBM WebSphere® Portal helps to improve employee productivity and strengthen relationships with customers and trading partners, by allowing users to interact in a personalized way with dynamic information, applications, processes, and people.

Some of the key features of the IBM WebSphere Portal offering are:

- It delivers a single, universal point of access that is integrated, highly customizable and scalable to interact with key applications, content, people, and business processes.
- It offers numerous portlets for e-mail, calendars, syndicated news, industry applications and many other functions.
- The WebSphere Portal *for Multiplatforms* includes three editions:
  - **Portal Enable** is the base offering, easily scalable to more advanced portal offerings as need may dictate.
  - **Portal Extend** features powerful collaborative, extended search and Web analysis features.
  - **Portal Experience** is a comprehensive solution for developing, deploying and maintaining enterprise portals.

This redbook covers the WebSphere Portal for z/OS and OS/390 product, which has portal solutions for enterprise customers needing the highest performance and reliability. At the time this redbook was written, the WebSphere Portal for z/OS and OS/390 product is based on the WebSphere Portal V4.1 Enable edition.

1.2 WebSphere Portal Enable

WebSphere Portal V4.1 Enable Edition is available on the z/OS and OS/390 platforms, and runs on WebSphere Application Server V4.1.4 for z/OS. The IBM WebSphere Portal Enable offering lets you quickly build highly scalable portals that simplify and speed up access to personalized information and applications. WebSphere Portal Enable provides common services including connectivity and integration that allows access to enterprise data, and other external applications.

1.2.1 WebSphere Portal for z/OS and OS/390 packages

Along with WebSphere Portal server, you get Personalization Server, WebSphere Portal Content Publishing (WPCP), WebSphere Transcoding

Publisher (WTP), User Management facility, security features, Click-to-Action (C2A), development tools and other features useful for constructing a portal site.

1.2.2 WebSphere Personalization Server

The WebSphere Personalization Server provides technologies for targeting Web content to meet user needs and preferences via rules-based personalization, where the business manager defines a set of business rules that determines whether the Web content is to be displayed for a particular user or a group of users.

1.2.3 WebSphere Portal Content Publishing

WebSphere Portal Content Publishing (WPCP) provides a browser-based interface that enables non-technical users to create, contribute, and manage content on portals and Web sites in a simple and controlled manner. Features like template management, workflow management, version control, and access control allow users to publish static or dynamic content quickly and efficiently. This provides portal end users with access to up-to-date information when they need it.

There are two parts to WPCP — the Publisher runtime and the Publisher Authortime. The WPCP Publisher runtime gets installed on the z/OS and OS/390 platforms, and the Authortime is provided for installation on a distributed platform. WPCP Authoring server uses a workflow that could be Lotus® Workflow™ V3.4 that requires Lotus Domino® components, or the stand-alone Lite Workflow that is provided with WPCP.

1.2.4 WebSphere Transcoding Publisher

This WebSphere Transcoding Publisher (WTP) component is installed along with WebSphere Portal Server. Portal users or administrators do not use or configure WTP directly, but the Portal Server uses it behind the scenes for the Web clipping feature and when the displayed portal pages need to be rendered for wireless devices using WML or cHTML.

1.2.5 WebSphere Studio Application Developer and the Portal Toolkit

WebSphere Portal comes with a tool for professional developers. It is called WebSphere Studio Application Developer (simply stated as *Application Developer*) and runs on the Windows platform as part of the portal development environment. It can be used for creating, testing, debugging, and deploying portlets, Java Server Pages (JSPs), servlets, EJBs and other assets related to portals and J2EE Web applications. The Portal Toolkit is provided as a plug-in to be installed within Application Developer to provide wizards and perspectives to assist with portlet development.

1.2.6 Summary

In terms of architecture and installation, WebSphere Portal Server, Personalization Server, WebSphere Transcoding Publisher, and the Publisher runtime for WebSphere Portal Content Publishing are installed on the z/OS and OS/390 platforms. This installation is covered in Chapter 2, “Portal installation” on page 19.

WPCP Authortime, and Application Developer typically gets installed on a Windows system or another supported distributed system and is covered in Chapter 3, “Development environment installation” on page 93.

The software components that make up WebSphere Portal are typically installed on hardware in production environments with two basic architectures — *two-tier* or t*hree-tier*. The hardware can be split up further creating n-tiers, but logically they map to two or three tier architectures using the Java 2 Platform, Enterprise Edition (J2EE) architecture as a reference. For information on J2EE and its architectures, refer to:

`http://java.sun.com/j2ee/overview.html`

1.3 Logical tiers

In these architectures, a tier is any component that participates in the actual application transaction. A Web server acting as a reverse proxy, a firewall, a switch or a router are not considered tiers. They are viewed as being part of the networking infrastructure.

Our view follows the J2EE application model with some variation. We have added a wireless client and a portlet takes the place of a servlet as shown in Figure 1-1 on page 5.

Portlets are reusable software components, running inside a portal, that provide access to Web-based content, applications, and other resources. From a user’s perspective, a portlet is a window on the portal’s Web browser screen, that provides a specific service or displays certain information. For people familiar with servlets in the Java world, a portlet is a specialized servlet. Portlets will be the focus of the architecture diagrams that follow and are covered in more detail in Chapter 6, “Portlet development and deployment” on page 181.

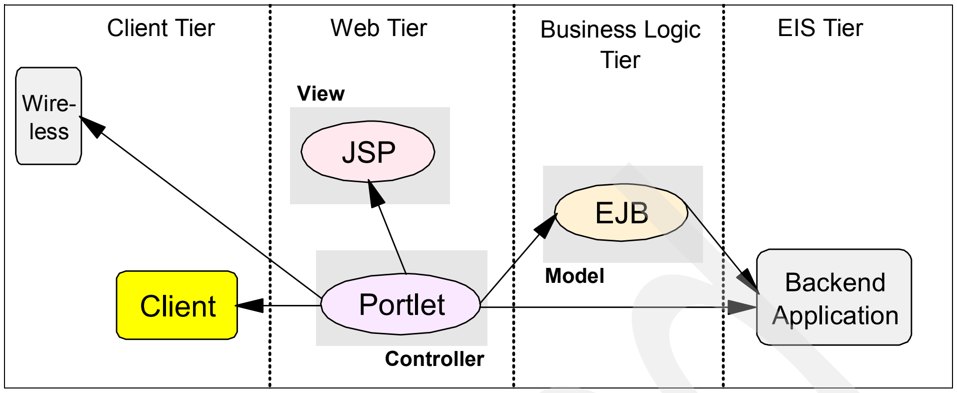


Figure 1-1 Tiers in the application model

- For the portal architecture, in the client tier there are no fat clients, only thin browser clients and wireless clients. It is possible to use Java applets in the client tier.
- The Web tier normally contains the Web server, presentation logic, and the controller functions handled by portlets. Note that a portlet is an extension of a servlet.
- The business logic tier will contain Enterprise Java Beans (EJBs).
- In the Enterprise Information System (EIS) tier you have backend applications, databases, and transaction systems.

1.4 Logical architectures

On the z/OS and OS/390 platform, the multi-tier J2EE architectures can be translated to two-tier, three-tier, or n-tier logical environments. We discuss the two-tier and three-tier architectures in this book.

1.4.1 Two-tier logical architecture

This is a simple architecture, where all the tiers except the client tier reside on the same server. It is relatively easy to configure and support and is shown in Figure 1-2 on page 6.

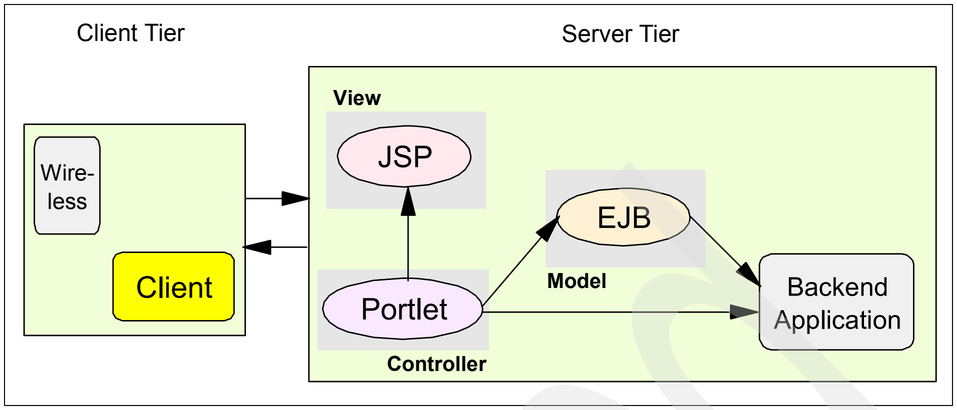


Figure 1-2 Two-tier logical architecture

For this redbook project, our test environment was set up as a two-tier architecture, with the WebSphere Application Server, WebSphere Portal Server, database server, and an LDAP server all installed on the server tier. A Web browser, the Authortime component of WebSphere Portal Content Publishing, and WebSphere Studio Application Developer was installed on a distributed platform, in this case two computers running Windows 2000.

1.4.2 Three-tier logical architecture

Implementing a three-tier architecture provides several options for a *logical view* of the architecture. The first option is to put the Web tier components (portlets, JSPs, and servlets) and EJB components on the same server as a middle tier, and the EIS components such as database and applications placed on a third or backend-tier. This is shown in Figure 1-3.

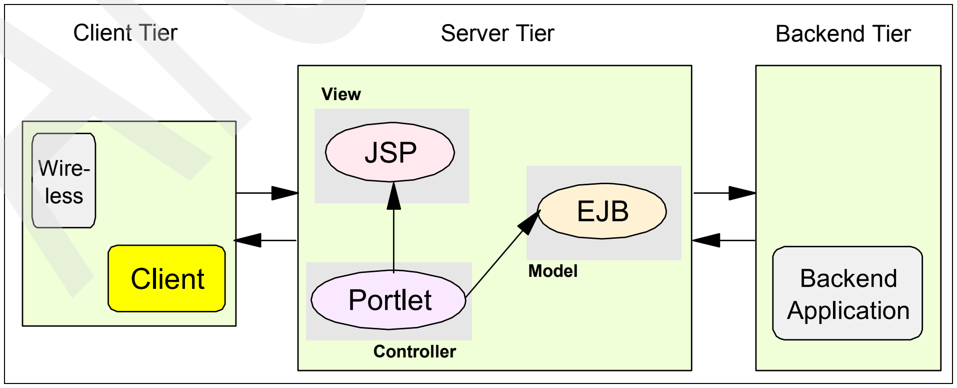


Figure 1-3 Three-tier logical architecture: Option one

Another option would be to keep the Web tier components (portlets, JSPs, and servlets) in the middle tier and co-locate the EJB components along with the EIS components (database and applications) on the back-end server. This option separates the business logic and data access from the presentation layer represented by portal server. This solution is sometimes referred to as the Distributed Application Model. On the zSeries platform, the middle tier and back-end tier could both reside on the same physical hardware, but still be completely separated as logical partitions or LPARs. Figure 1-4 shows option two in a three-tier architecture.

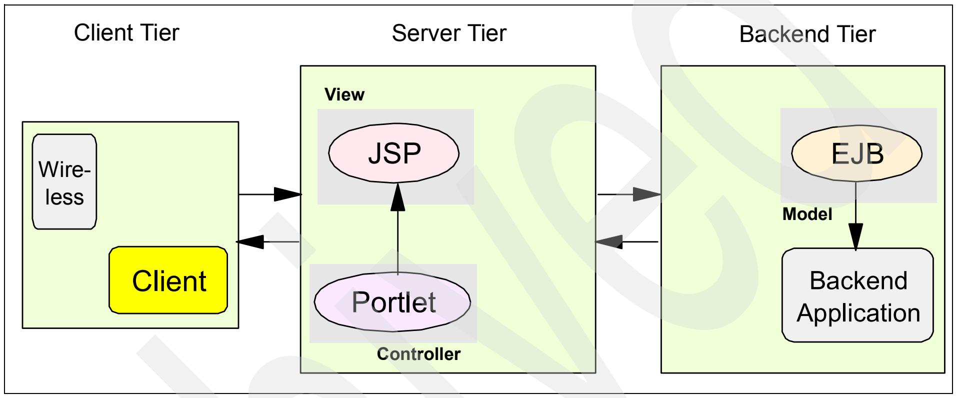


Figure 1-4 Three-tier logical architecture: Option two

1.5 Physical architectures

Since the different components required in a portal installation can exist separately or be co-located, there are various *physical* architectures that can be implemented. The driving factors would be cost, and Quality of Service (QoS). Quality of Service is further defined by availability, security, maintainability, reliability, and service levels.

WebSphere Portal Server requires WebSphere Application Server and is configured like any other J2EE Server on zSeries with a Control Region and one or more Server Regions. Personalization server, WebSphere Portal Transcoding Publisher, and WebSphere Portal Content Publishing Runtime are the other software components that get added to the WebSphere Portal server J2EE application.

1.5.1 Two-tier physical architecture

With portal server acting as the presentation layer, the two-tier architecture is rather easy to set up. Figure 1-5 shows the architecture containing one Control Region (CR) and a Server Region (SR). Client portal requests arrive at the HTTP Transport Handler (HTH) and are passed on to WebSphere Portal server. If needed the backend application tier is accessed and the results returned to the Web browser.

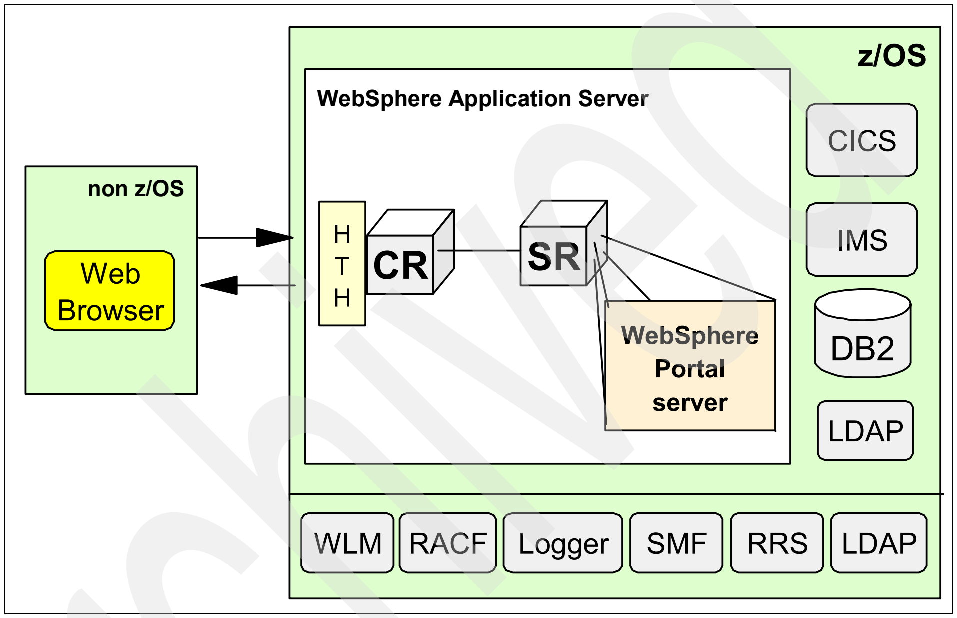


Figure 1-5 Basic two-tier configuration

We do not recommend that you use the LDAP server instance that is used by WebSphere Application Server. Instead, create another instance of the base LDAP server for the Portal server, because of potential issues with access contention. Even though the LDAP server can exist anywhere on the z/OS and OS/390 system, we recommend that it reside on the same image as the WebSphere Portal server.

For a description of the rest of the usual z/OS and OS/390 components, refer to *e-business Cookbook for z/OS: Technology Introduction,* SG24-5664.

The z/OS and OS/390 platforms come with Workload Manager (WLM) giving you inherent workload distribution and load balancing. Figure 1-6 on page 9 shows how WLM could start up Server Regions either via a parameter during setup or on its own based on system load. This architecture starts to address scalability but not high availability.

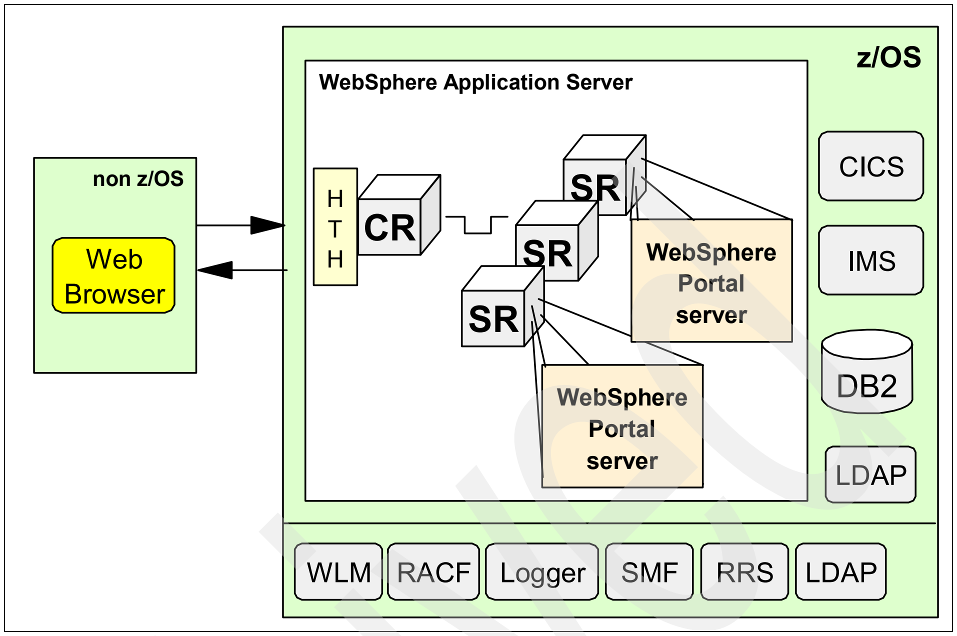


Figure 1-6 Two-tier architecture with scalability

Chapter 2, "Portal installation" on page 19 discusses how multiple Server Regions were created with the same Control Region in our test environment. In our simple test the WLM facility was able to serve 100 plus concurrent users with a good response characteristics using a WebSphere Workload Simulator. Certainly this small test did not replicate any production customer scenario.

1.5.2 Three-tier physical architecture

In many cases, the a WebSphere J2EE application is used to access existing data on a production z/OS system. The middle-tier could have WebSphere Application Server on a z/OS or non-z/OS system. If a z/OS system is used for both the middle tier and the back-end tier, and they are both in the same parallel sysplex, then the benefits include better security and higher Quality of Service. A sample three-tier architecture is shown in Figure 1-7 on page 10.

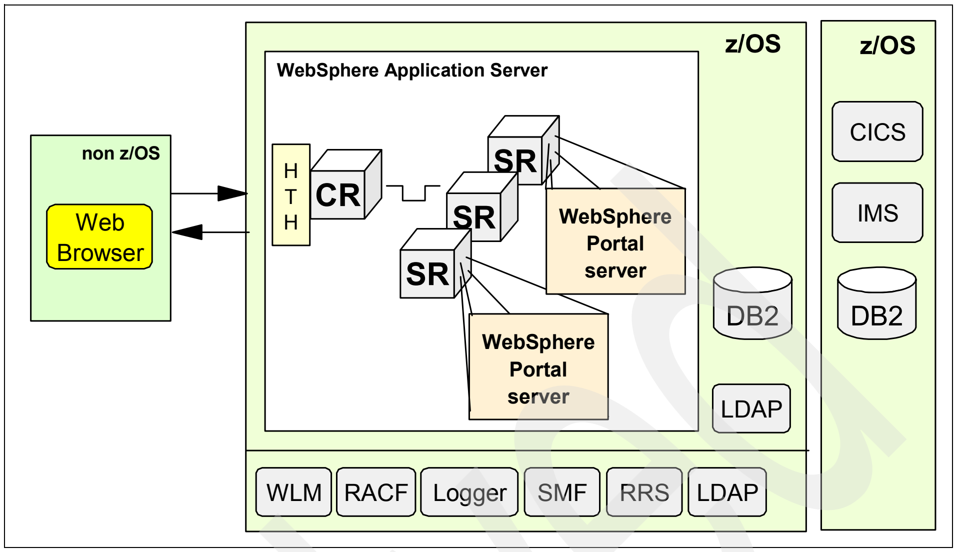


Figure 1-7 Sample three-tier architecture

As far as WebSphere Portal is concerned there is really not much difference between the two-tier and three-tier architectures, because everything required by WebSphere Portal is found on tier one and tier two. WebSphere Application Server repository has to exist on the same system. Only production-related database and other back-ends would be in the third-tier. Custom portlets accessing those third-tier back-end systems would be used within WebSphere Portal server to get information to the portal user.

1.6 Our architecture

During this redbook project we set up two architectures. Our initial architecture was simple to enable the installation efforts. Later we switched on other components to get closer to real customer environments.

1.6.1 Our initial architecture

For this redbook, we started by keeping things simple and used the two-tier architecture with a single WebSphere Application Server Control Region (CR) and one Portal Server Region (SR). We also felt that, in the early stages of deployment and test, this would be the most common option for implementation by IBM customers. The architecture shown in Figure 1-8 on page 11 does not explicitly address scalability, but during the WebSphere Portal installation there is a parameter that controls the number of Server Regions to create. We left that parameter to unlimited, giving WLM the ability to generate Server Regions if the need arose. Therefore, the second SR and WebSphere Portal server are shown in the figure but are not highlighted in bold.

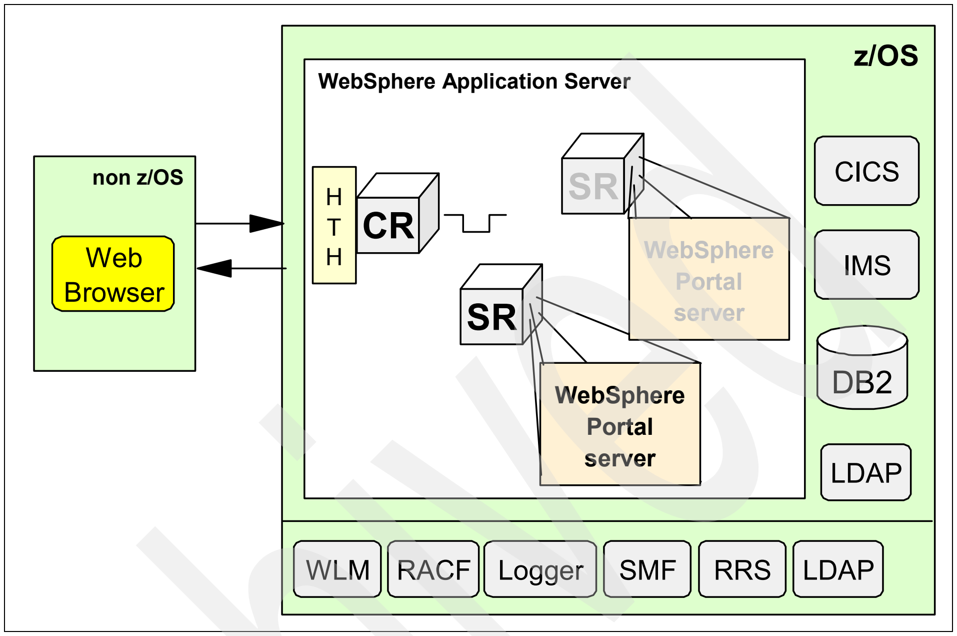


Figure 1-8 Our initial WebSphere Portal architecture

Installation summary

WebSphere Portal Server was installed by creating another J2EE server on WebSphere Application Server on z/OS or OS/390. Then the administrative portlets were installed. After verifying that portal server was up and running the additional portal components, namely WebSphere Transcoding Publisher (WTP), WebSphere Portal Content Publishing (WPCP) Publisher runtime, and Personalization, were added. The Personalization component actually gets installed as part of WPCP.

Figure 1-9 on page 12 gives you a glimpse inside WebSphere Portal Server. Therefore, whenever we show or discuss the WebSphere Portal Server it implies that it also contains WTP, WPCP, and the administrative portlets.

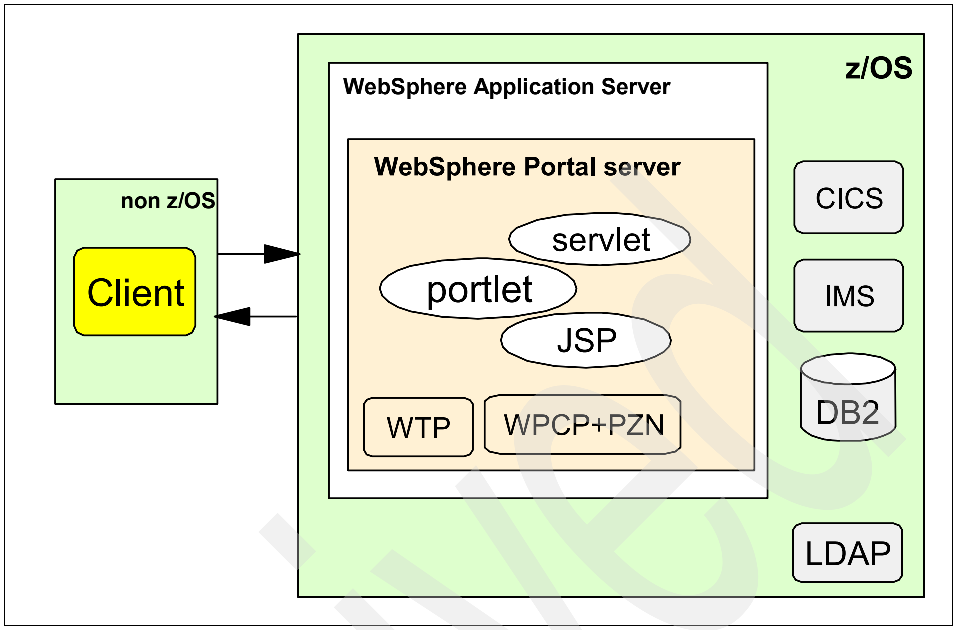


Figure 1-9 Inside WebSphere Portal on z/OS and OS/390

1.6.2 Our preferred architecture

In the architecture as shown in Figure 1-8 on page 11 the incoming HTTP request is handled by the HTTP Transport Handler (HTH) via port 8082. This is one of the ways of testing browser access into WebSphere. We do not recommend this in a production environment, because you are directly allowing access to a port in WebSphere Application Server's Control Region.

The preferred way to handle portal access is by having an IBM HTTP Server on the z/OS and OS/390 machine that is accessible to clients via port 80. The IBM HTTP Server HTTP plug-in then sends the request across to the HTTP Transport Handler that is listening on port 8082. We set this up and tested the portal functionality for our environment. Figure 1-10 on page 13 shows the recommended architecture.

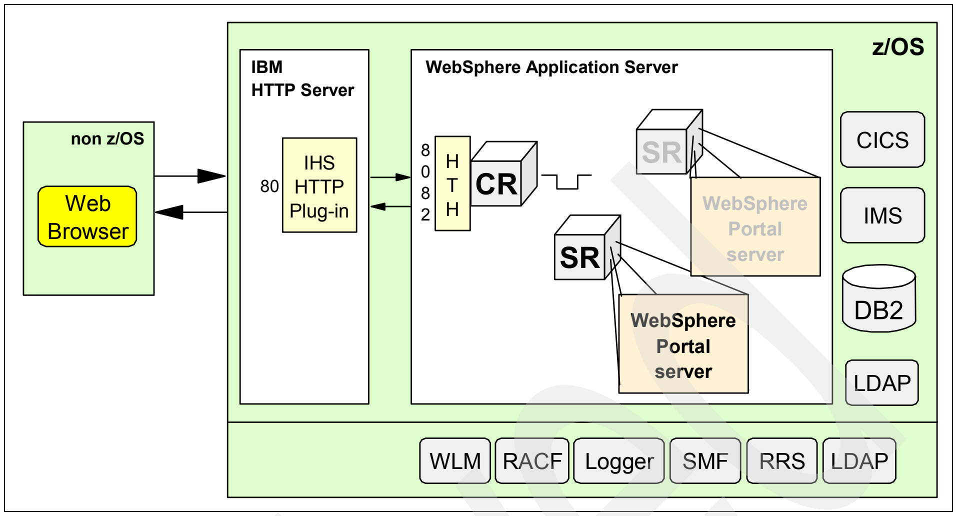


Figure 1-10 Our preferred architecture

1.6.3 Our overall environment

Table 1-1 lists the major software products and release levels that we used. You will find a detailed description of all the software components that were used in Chapter 2, "Portal installation" on page 19.

Table 1-1 OS and other software product levels used in this project

Product name	Release level
z/OS	V1.3
DB2®	V7.1
IBM HTTP Server for z/OS	V1.3.19
WebSphere Application Server	V4.1
WebSphere Portal server	V4.1
WebSphere Transcoding Publisher	V4.1
WebSphere Portal Content Publishing	V4.2
z/OS LDAP Server	V1.2

Our test environment was set up in this manner:

- On the zSeries system, running z/OS V1.3, we installed the main portal components — WebSphere Portal server, WebSphere Portal Content

Publishing Runtime Server, WebSphere Transcoding Publisher, and Administrative portlets. Note, our system already had a DB2 database server, native LDAP server, and WebSphere Application Server. For portal purposes we created another instance of the LDAP server which is shown as two LDAP boxes in Figure 1-11 on page 15.

- On a PC, running Windows 2000 SP2®, we installed a Web browser and the Authortime component of WebSphere Portal Content Publishing.
- On another PC, running Windows 2000 SP2, the development tooling was installed, that is, WebSphere Studio Application Developer V4.1 with plug-ins for the Portal Toolkit and WPCP wizards.

For detailed information on our installation and setup refer to Chapter 2, "Portal installation" on page 19 and Chapter 3, "Development environment installation" on page 93.

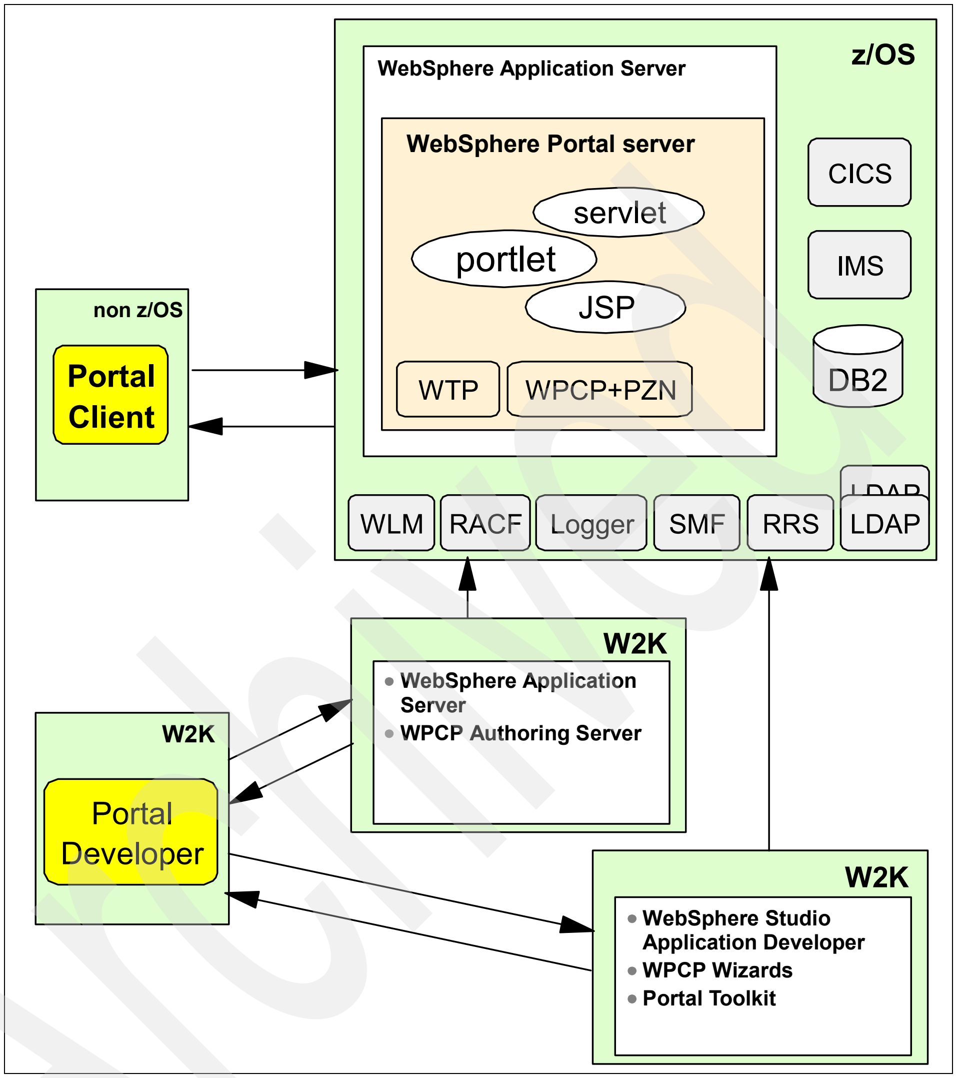


Figure 1-11 Our WebSphere Portal environment

The Portal environment on zSeries lends itself well to best practices or application patterns that make up the *Portal Composite Pattern*. These Application Patterns are:

- Access Integration::Web Single Sign-On Application Pattern
- Access Integration::Pervasive Device Access Application Pattern
- Access Integration::Personalized Delivery Application Pattern

- Self Service::Directly Integrated Single Channel Application Pattern

These and other patterns are detailed in the redbook *A Portal Composite Pattern Using WebSphere V4.1,* SG24-6869.

1.7 Benefits of having WebSphere Portal on z/OS

So far we have described how the WebSphere platform can be utilized on two-tier and three-tier architectures and the benefits it derives from the J2EE implementation and the Model-View-Controller (MVC) paradigm. Using the zSeries platform we can get more advantages by using its work load management (WLM) features, high-speed connect facilities, and the attributes that provide extremely high levels of availability and QoS to users.

On a distributed platform we would recommend, to our WebSphere Portal customers, that you create three environments — development, test, and production — with the test environment mirroring the proposed production environment. That normally involves, at a minimum, three different physical systems and each system would have one or more hardware boxes. On the z/OS platform we can take advantage of the power of the z/OS architecture and install three totally different instances of WebSphere Portal in the same logical partition (LPAR) or different LPARs using the same physical hardware platform.

Scalability on the distributed platform is achieved either by vertical scaling or horizontal scaling. Vertical scaling means creating portal server clones on the same machine, and horizontal scaling involves creating portal server clones on different machines that are part of the same WebSphere domain. On the z/OS platform, it simply involves specifying the minimum and maximum Server Region (SR) values. Then the zSeries WLM feature handles scaling based on the type of load and number of users that access the portal.

Another common requirement for our Portal customers is that they would like to separate Intranet, Internet, and Extranet users of the portal. This is required from a Quality of Service (QoS) perspective, and also to create portals that each have a different look and feel. Again, this can be done on the z/OS platform, because the portals can have totally different instances on different WebSphere instances in a sysplex.

Note: We did not set up and test this scenario.

There are additional benefits of installing the LDAP server, used by WebSphere Portal, on z/OS:

- First is availability. LDAP server, if installed and configured in a z/OS Parallel Sysplex with shared DB2 and Sysplex Distributor, provides a highly available, scalable LDAP server that can be used by any client on the network.
- The second is *native authentication*. Native authentication allows connection between the LDAP server and Resource Access Control Facility (RACF) wherein the userid and password that is used to authenticate to LDAP is actually passed to the Security Access Facility (SAF) to be verified. This setup could be used to allow Internet portal customers to authenticate directly against the LDAP server, while Intranet users who already had a userid on the z/OS system, would authenticate using their RACF userid and password.

As we gain more experience with WebSphere Portal on z/OS and OS/390 we are sure there will be other scenarios which we will discover that will not only highlight the benefits of running WebSphere Portal on z/OS, but also of running different architectures.

2

Portal installation

This chapter describes the installation and post-installation steps of WebSphere Portal Server on z/OS.

The installation of WebSphere Portal Server uses SMP/E, followed by a series of post-installation steps that uses pre-configured jobs to perform customization of the SMP/E installation.

The installation process described here, reflects our experiences performing the Portal installation and configuration of the following software components:

- WebSphere Portal Server
- WebSphere Transcoding Publisher
- WebSphere Portal Content Publishing

2.1 System prerequisites

In this section we list the hardware and software prerequisites for WPS.

2.1.1 Hardware prerequisites

The hardware prerequisites are:

- zSeries, 9672 G5 or later, MP3000, IEEE floating point hardware
- One or more cryptographic co-processors and/or PCI cards (if using SSL)
- IP network attachments (OSA) to support clients/users load

2.1.2 Software prerequisites

The software prerequisites are:

- OS/390 V2R10, z/OS V1R2 or higher
- WebSphere Application Server for z/OS V4.01 at Service Level 401403 or later
- IBM Security Server (RACF or equivalent)
- LDAP Server
- DB2 V7.1 (JDBC 2.0 driver)
- Java SDK 1.3.1: J2RE 1.3.1 IBM OS/390 Persistent Reusable VM build `hm131s-20021114a` or better
- Unicode support
  - OS/390 V2R10 requires HUNI2A0 (ordered separately)
  - z/OS V1R2 and V1R3 requires HUN7705 (shipped with z/OS)

  Attention: Unicode code support should be installed and configured as part of the base system preparation. This is due to the fact that the Unicode conversion environment is initialized at IPL and therefore, an IPL is required to activate the support. See 2.4.3, “UNICODE support” on page 27.

- FTP Server

2.1.3 Our environment

The environment we used for this redbook project, was intended to be representative of a setup that customers could use for testing WebSphere Portal and performing a Proof of Concept of the Portal functions. This was a brand new installation of Portal Server, and not a migration from PTF1 to PTF2. Our set up is reflected in the following system environment.

z/OS environment

Our z/OS environment includes:

- z/OS V1R3
- WebSphere Application Server V4.01 at Service Level W401503
- SDK 1.3.1 at Service Release 20030204
- LDAP IBM Directory Server V1.2
- DB2 V7.1, JDBC 2.0
- TCP/IP environment
- Windows 2000, service pack 3

Note: For migration from PTF1 to PTF2, please see the README file at:

`http://publib.boulder.ibm.com/pvc/wp/415/zos/en/readme41ptf2_zos.html`

2.2 What is in the box

First of all, make sure you have the complete product package and documentation as delivered.

2.2.1 SMP/E tape for install on z/OS

The tape contains tar files for the following components:

- WebSphere Portal Server Enable
- WebSphere Transcoding Publisher
- WPCP (WebSphere Portal Content Publishing run-time)

2.2.2 CD-ROMs for distributed platform

These are the CD-ROMs for the distributed platform:

- WebSphere Portal Enable for multiplatforms
- WebSphere Application Server for multiplatforms
  - Single Server (for the WebSphere Studio Application Developer)
  - Advanced Edition for WPCP

- WebSphere Portal Content Publishing and WebSphere Personalization
- Lotus Domino Application Server
- DB2 UDB
- Secureway Directory Server (LDAP)
- IBM HTTP Server (IHS)
- WebSphere Studio Application Developer (Application Developer)
- Portal Toolkit
- Product Fixpacks

Note: The WebSphere Application Server for multi-platforms, and WebSphere Portal Enable for multi-platforms, are *not* licensed for production use.

Important: Refer to Chapter 3, "Development environment installation" for details about the components shipped on the CD-ROMs.

2.3 Preparing to install

Before installing the product, you should review the PSP bucket by using the following values to locate relevant items:

```
Upgrade: HSPORTAL41
Subset: HPSV413
Subset: JPSV413
Subset: JPSV414
```

2.4 SMP/E installation

WebSphere Portal for z/OS consists of WebSphere Portal Server Enable, WebSphere Transcoding Publisher, and WebSphere Portal Content Publishing. These products are delivered as an SMP/E installable component, each with its own separate FMID and COMPID as shown in Table 2-1.

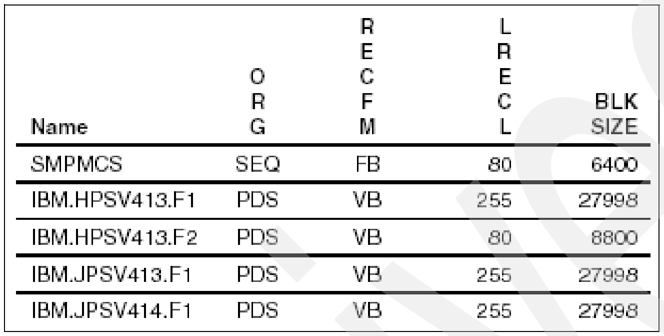
Table 2-1 FMIDs and COMPIDs

FMID	COMPID	Component name
HPSV413	5655K1200	Portal server
JPSV413	5655K1201	Transcoding publisher

FMID	COMPID	Component name
JPSV414	5655K1202	Content publisher

SMP/E content description

WebSphere Portal for z/OS is packaged on a 3480 cartridge. The feature number is 5802 and is contained in one volume, with a VOLSER of PSV413. Figure 2-1 describes the program file content for Portal for z/OS.



Name	ORG	RECFM	LRECL	BLK SIZE
SMPMCS	SEQ	FB	80	6400
IBM.HPSV413.F1	PDS	VB	255	27998
IBM.HPSV413.F2	PDS	VB	80	8800
IBM.JPSV413.F1	PDS	VB	255	27998
IBM.JPSV414.F1	PDS	VB	255	27998

Figure 2-1 Program file content

2.4.1 WebSphere Portal Server Enable offering for z/OS

WebSphere Portal Server for z/OS is the Enable version of the WebSphere Business Portal offerings that contains three modules as shown in Figure 2-2 on page 24:

- WebSphere Portal Server
- WebSphere Personalization
- WebSphere Portal Content Publishing

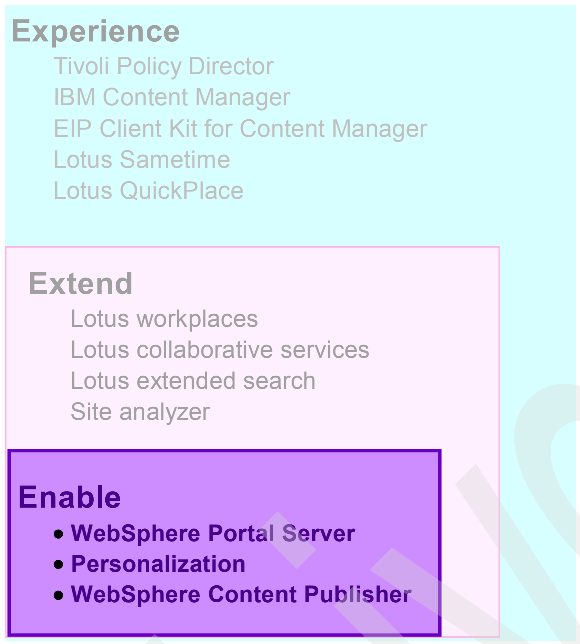


Figure 2-2 WebSphere Portal Servers family

2.4.2 Prerequisites

The following list of software prerequisites must be installed and operational before proceeding with the Portal Server installation:

- WebSphere Application Server must be current up to level UQ75293 (W401502), including all prerequisites.
- The WebSphere Application Server installation must be verified by running the installation verification BBOIVPE.
- A configured LDAP Server must be accessible.
- The following Java version must be used: J2RE 1.3.1 IBM OS/390 Persistent Reusable VM build `hm 131s-20021114a`.
- JDBC driver maintenance forward to PQ61146 and PTF UQ73041.

Maintenance considerations

Before performing the SMP/E installation steps, if you want to install Portal for z/OS to a different HFS than the default root HFS, you will need to create a new HFS file system. This new HFS has to be at least 600 cylinders in size as described in HOLD UA01959.

For our redbook project environment we decided to handle different versions of Portal for z/OS installations. To do that we used a /SERVICE directory to perform installation and maintenance of the application, and the default directory of /usr/lpp/PortalServer directory to install the current version of the product. We encourage you to do the same, so that you can easily maintain your Portal for z/OS installation.

In our case we used the following:

- WPS.SERVICE.HFS was the file system mounted from /SERVICE/usr/lpp/PortalServer, where we performed installations and applied maintenance.
- WPSPROD.HFS was the file system mounted from /usr/lpp/PortalServer where we ran Portal for z/OS.

Note: The portal installation changes its own HFS and also modifies WebSphere for z/OS HFS files through SM API. Make sure that you follow the maintenance installation instructions and not just modifying the portal HFS files.

Important: Ensure that the HFS file containing /tmp directory has at least 30 MB free space.

SMP/E RECEIVE Jobs

Here are the SMP/E jobs used for the portal residency; you may need to customize the receive jobs so that they fit your specific environment:

Portal product SMP/E job

`SMPRDAS1`: This job does the receive of the Portal product and is shown in Example 2-1.

Example 2-1 Receive Portal Product

```
//SMPRDAS1 JOB (999,POK),'SMPEREC',CLASS=A,REGION=0M,
//             MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*****************************************************************
//**  THIS JOB DOES THE RECEIVE OF THE PORTAL PRODUCT FROM
//*   RELFILES ON DASD.
//*****************************************************************
//SMP      EXEC PGM=GIMSMP,REGION=0K,       SMPE
//             PARM='',
//             TIME=1000
//*****************************************************************
//*        SYSOUT DATA SETS
//*****************************************************************
//SMPPTFIN DD DISP=SHR,DSN=JAVA1.EJP.HPSV413.SMPMCS
//         DD DISP=SHR,DSN=JAVA1.EJP.JPSV413.SMPMCS
//         DD DISP=SHR,DSN=JAVA1.EJP.JPSV414.SMPMCS
//SMPLOG   DD SYSOUT=*
//SMPLIST  DD SYSOUT=*
//SMPOUT   DD SYSOUT=*
//SMPRPT   DD SYSOUT=*
//SMPHOLD  DD DUMMY
//SYSUT1   DD DSN=&UT1,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//SYSUT2   DD DSN=&UT2,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//SYSUT3   DD DSN=&UT3,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//SYSUT4   DD DSN=&UT4,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//******************************************************************
//*      SMP/E   DATA SETS
//******************************************************************
//SMPCSI   DD DISP=SHR,DSN=WAS41.SMPE.CSI,
//         AMP='BUFNI=20,BUFND=20'
//*
//SMPCNTL  DD *
 SET BDY(GLOBAL) .
 RECEIVE SELECT(HPSV413
               JPSV413
               JPSV414)
   SYSMODS LIST RFPREFIX(JAVA1) .
/*
//
```

Portal product maintenance SMP/E job

`SMPRDAS2`: This job does the receive of the Portal product PTFs and is shown in Example 2-2.

Example 2-2 Receive Portal Maintenance PTFs

```
//SMPRDAS2 JOB (999,POK),'SMPEREC',CLASS=A,REGION=0M,
//             MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//******************************************************************
//**  THIS JOB DOES THE RECEIVE OF THE PORTAL PRODUCT MAINTENANCE PTFs
//*   FROM RELFILES ON DASD.
*******************************************************************
//SMP      EXEC PGM=GIMSMP,REGION=0K,       SMPE
//             PARM='',
//             TIME=1000
//******************************************************************
//*        SYSOUT DATA SETS
//******************************************************************
//SMPPTFIN DD DISP=SHR,DSN=JAVA1.UA01959
```

```
//         DD DISP=SHR,DSN=JAVA1.UA01969
//         DD DISP=SHR,DSN=JAVA1.UA01970
//         DD DISP=SHR,DSN=JAVA1.UW95968
//         DD DISP=SHR,DSN=JAVA1.UW95969
//         DD DISP=SHR,DSN=JAVA1.UW95970
//SMPLOG   DD SYSOUT=*
//SMPLIST  DD SYSOUT=*
//SMPOUT   DD SYSOUT=*
//SMPRPT   DD SYSOUT=*
//SMPHOLD  DD DUMMY
//SYSUT1   DD DSN=&UT1,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//SYSUT2   DD DSN=&UT2,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//SYSUT3   DD DSN=&UT3,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//SYSUT4   DD DSN=&UT4,DISP=(NEW,PASS),UNIT=3390,
//            SPACE=(CYL,(40,10)),VOL=SER=TARTS1
//*****************************************************************
//*      SMP/E   DATA SETS
//*****************************************************************
//SMPCSI   DD DISP=SHR,DSN=WAS41.SMPE.CSI,
//         AMP='BUFNI=20,BUFND=20'
//*
//SMPCNTL  DD *
 SET BDY(GLOBAL) .
 RECEIVE SYSMODS LIST .
```

Portal Product PTFs

Since this was a new installation we needed to install the Portal product base plus both PTFs:

- PTF2 includes JAVA1.UA01959, JAVA1.UA01969, JAVA1.UA01970
- PTF1 includes JAVA1.UW95968, JAVA1.UW95969, JAVA1.UW95970

2.4.3 UNICODE support

Before installing Portal for z/OS, you should ensure the system has UNICODE support installed. You can easily check for that by issuing TSO command (Example 2-3):

```
/D UNI,ALL
```

Example 2-3 Checking for UNICODE support

```
D UNI,ALL
 CUN3000I 18.53.15 UNI DISPLAY 301
  ENVIRONMENT: CREATED       04/29/2003 AT 15.41.39
```

```
          MODIFIED      04/29/2003 AT 15.41.40
          IMAGE CREATED 04/29/2003 AT 15.12.57
 SERVICE: CUNMCNV   CUNMCASE
 STORAGE: ACTIVE       66 PAGES
          LIMIT     51200 PAGES
CASECONV: NORMAL
CONVERSION: 00850-01047-ER            01047-00850-ER
```

If you don't get the result shown in Example 2-3 on page 27 then you need to activate the conversion tables.

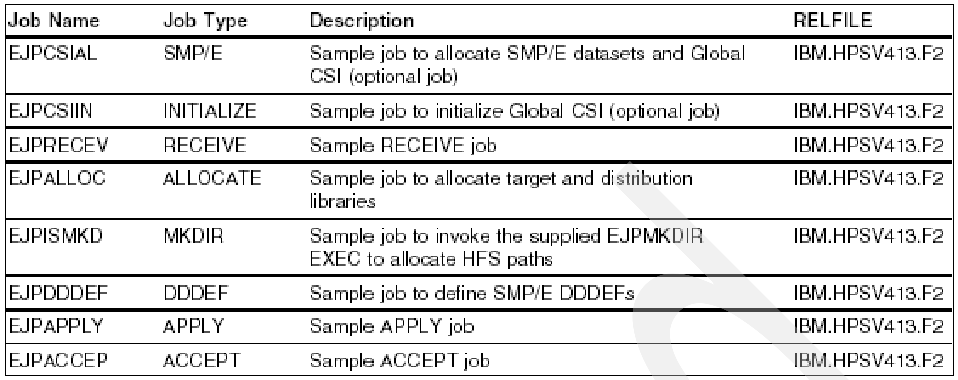
You will find a sample job for activating the conversion tables in Example 2-4.

Example 2-4 Conversion table sample job

```
//CUNIMAGE JOB (POK,999),'UNICODE',MSGLEVEL=(1,1),MSGCLASS=T,
//  CLASS=A,NOTIFY=&SYSUID
//*******************************************************************
//* IMAGE GENERATOR                                                 *
//*******************************************************************
//CUNMIUTL EXEC PGM=CUNMIUTL
//STEPLIB  DD   DSN=SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//TABIN    DD   DISP=SHR,DSN=SYS1.SCUNTBL
//SYSIMG   DD   DSN=SYS1.PARMLIB(CUNIMG00),DISP=SHR
//SYSIN    DD   *
  /*******************************************
   * INPUT STATEMENTS FOR THE IMAGE GENERATOR *
   *******************************************/

     CASE NORMAL;              /* ENABLE TOUPPER AND TOLOWER */
     CONVERSION 00850,01047,ER; /*ASCII       -> EBCDIC     */
     CONVERSION 01047,00850,ER; /*EBCDIC      -> ASCII      */
/*
//
```

2.4.4 Performing the SMP/E installation

Portal for z/OS SMP/E installation comes with a set of jobs to perform the RECEIVE, APPLY, and ACCEPT commands. These jobs are located in the IBM.HPSV413.F2 dataset as described by Figure 2-3 on page 29.



Job Name	Job Type	Description	RELFILE
EJPCSIAL	SMP/E	Sample job to allocate SMP/E datasets and Global CSI (optional job)	IBM.HPSV413.F2
EJPCSIIN	INITIALIZE	Sample job to initialize Global CSI (optional job)	IBM.HPSV413.F2
EJPRECEV	RECEIVE	Sample RECEIVE job	IBM.HPSV413.F2
EJPALLOC	ALLOCATE	Sample job to allocate target and distribution libraries	IBM.HPSV413.F2
EJPISMKD	MKDIR	Sample job to invoke the supplied EJPMKDIR EXEC to allocate HFS paths	IBM.HPSV413.F2
EJPDDDEF	DDDEF	Sample job to define SMP/E DDDEFs	IBM.HPSV413.F2
EJPAPPLY	APPLY	Sample APPLY job	IBM.HPSV413.F2
EJPACCEP	ACCEPT	Sample ACCEPT job	IBM.HPSV413.F2

Figure 2-3 List of installation jobs

Before modifying these jobs we copied them to a TSO JAVA1 userid JCL dataset called `JAVA1.JCL`.

SMP zone considerations

We decided to use the zone of WebSphere Application Server for z/OS to perform Portal for z/OS SMP/E installation. As dependencies existed only on the Application Server we considered this to be a good solution.

Before submitting the jobs, check to make sure that the following three F1 datasets are in the catalog. In a standard installation that would mean that the following datasets have to be cataloged:

- IBM.HPSV413.F1
- IBM.JPSV413.F1
- IBM.JPSV414.F1

In our case the following datasets had been added to the catalog before performing the SMP/E RECEIVE:

- JAVA1.EJP.HPSV413.F1
- JAVA1.EJP.JPSV413.F1
- JAVA1.EJP.JPSV414.F1

Allocate SMP/E target

Customize the allocation job `EJPALLOC` to make it match your specific requirements, for example: HLQ, VOLumes, and so on, as shown in Example 2-5 taken from our system:

Example 2-5 Allocate target

```
(...)
//ALLOCT EXEC ALLOCTGT,
//             HLQ=WPS,      * EJP is the default
//             DSP=CATLG,    * CATLG is the default
//             TVOL1=WASTV1  * No default; volume1 for target libs
//*
//ALLOCD EXEC ALLOCDLB,
//             HLQ=WPS,      * EJP is the default
//             DSP=CATLG,    * CATLG is the default
//             DVOL=WASTV2   * No default; volume for dist. libs
//*
(...)
```

Create structure in target HFS

As mentioned in “Maintenance considerations” on page 24 we created a specific file system with a mountpoint to perform installation and apply maintenance. So the `EJPISMKD` job had to be tailored accordingly. Example 2-6 shows how we adapted the job for the maintenance process.

Example 2-6 Create HFS structure

```
//IKJEFT1A EXEC PGM=IKJEFT1A
//SYSEXEC DD DISP=SHR,
//      DSN=WAS41.HPSV413.F2
//*
//*SYSEXEC DD UNIT=SYSALLDA,VOL=SER=ttttt2,DISP=SHR,
//*      DSN=EJP.AEJPSAMP
//*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
PROF MSGID
EJPMKDIR /SERVICE
```

DDDEF entries

Now, customize job `EJPDDDEF` to make it match your specific requirements. Keep in mind that if you have chosen to follow our maintenance process you will have to tailor that job too, as shown in Example 2-7.

Example 2-7 Customize maintenance directory

```
(...)
//DEFPATH EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI  DD   DSN=WAS41.SMPE.CSI,
//             DISP=SHR
//SMPCNTL DD *
    SET BDY(WAST) .           /* change -PathPrefix- */
      ZONEEDIT DDDEF.
      CHANGE PATH('/usr/lpp/PortalServer'*,
                  '/SERVICE/usr/lpp/PortalServer'*).
    ENDZONEEDIT.
/*
```

Perform SMP/E APPLY and ACCEPT

Customize the APPLY and ACCEPT jobs, `EJPAPPLY` and `EJPACCEP`, to match your installation requirements. You can bypass the HODLDSYSTEM and get a return code of 0.

2.4.5 Post SMP/E installation

The post SMP/E installation steps turn out to be the real installation steps that you would normally have to do on a WebSphere Portal distributed platform.

Note: The default path of WebSphere Portal, #WPS# for our installation, refers to /usr/lpp/PortalServer, and the #DRVNUM# refers to the driver number 4.1.2. You may use other values for #WPS# for your installation.

After the SMP/E installation you should have six files in the directory you chose as the base #WPS# path. Table 2-2 shows the six files you should have. The three non version-suffixed files are the WebSphere Portal Server for z/OS three components at PTF1 level. The three files with 4.1.2 suffix are the PTF2 level of the three components.

Table 2-2 post SMP/E HFS files output result

File name	File size	Component
ejppsrv.tar	40826880	Portal Server base PTF1
ejppsrv.4.1.2.tar	24432640	Portal Server base PTF2
ejpwcpub.tar	11028480	Content Publishing PTF1
ejpwcpub.4.1.2.tar	3317760	Content Publishing PTF2
ejppsrv.tar	40826880	Transcoding Publish. PTF1
ejppsrv.4.1.2.tar	24432640	Transcoding Publish. PTF2

The actual process of installing PTF2 from scratch is different for each of the components as follows:

- Portal Server base: PTF2 is an update to PTF1
- Content Publishing: PTF2 is an update to PTF1
- Transcoding Publisher: PTF2 is a full product replacement

This basically means that the next step will prepare installation for:

- Portal Server base at PTF1 level full product
- Portal Server base PTF2 level
- Content Publishing at PTF1 level full product
- Content Publishing PTF2 level
- Transcoding Publisher at PTF2 level full product

2.5 Configure and install WebSphere Portal Server

After completing the SMP/E steps, you are ready to do the actual Portal server installation. This is a twelve step process. The residency portal installation process was based on the README *Installing WebSphere Portal on z/OS and OS/390 Version 4.1 PTF2*. This README file can be found at:

`http://publib.boulder.ibm.com/pvc/wp/415/zos/en/readme41ptf2_install_zos.html`

We will describe our installation for each of these steps:

- Check that the base z/OS, OS/390 operating system is up and running.
- WebSphere Application Server V4.01, LDAP and DB2 should be up and running.

Common step for all three Portal components

You will be extracting the tar files mentioned in Table 2-2 on page 31 for all Portal components.

Note that since Transcoding Publisher is a full product replacement, we will not extract the PTF1 level product tar file, but will directly extract the PTF2 level tar file.

The SMP/E process will have created a /usr/lpp/PortalServer directory either in the /SERVICE directory or in the root according to your customization of job `EJPDDDEF`.

2.5.1 Step one: Read the package contents

Important: Here we introduce a notation <WPS_PATH> that we use in this chapter and refers to the directory: /usr/lpp/PortalServer/PortalServer. This is the same as WPS_PATH environment variable.

In other words, this notation refers to #WPS#/PortalServer.

The installation file `ejppsrv.tar` is installed by SMP/E. This file contains the following directories and files:

- `wsport` directory is located at <WPS_PATH>/zosinst and contains reference files as follows:
  - `current.env`
  - `jvm.properties`
  - `trace.dat`
  - `webcontainer.conf`
- `jcl` directory is located at <WPS_PATH>/zosinst and contains:
  - Procedures to create and fill the DB2 database tables
  - JCL procedures for RACF setup
  - Control and server region procedures for WebSphere Portal
  - JCL procedures to set up WebSphere Portal on WebSphere Application Server for z/OS or OS/390
- `ear` directory is located at WPS_PATH and contains files:
  - `wsproxy.ear`
  - `wps.ear`
- `scripts` directory is located at <WPS_PATH>/zosinst and contains REXX and shell scripts to set up WebSphere Portal on WebSphere Application Server for z/OS or OS/390. These scripts are called by JCL procedures located in the `jcl` directory.
- `tools` directory is located at <WPS_PATH>/zosinst and contains the `390fy` tool and deployment scripts used during the installation of base portlets.
- `app` directory is located at <WPS_PATH> and contains the base portlets.

2.5.2 Step two: Unpack the product files

To unpack the product files

1. Log on to the UNIX System Services (USS) on the target system as a user with root authority, UID=0).
2. Change directory to #WPS#
3. Execute commands:

```
tar -xv -f ejppsrv.tar
tar -xv -f ejpwcpub.tar
```

To unpack the PTF2 product files

1. Log on to the UNIX System Services on the target system as a user with root authority.
2. Change directory to #WPS#
3. Execute commands:

```
tar -xv -f ejppsrv.4.1.2.tar
tar -xv -f ejpwcpub.4.1.2.tar
tar -xv -f ejptrcod.4.1.2.tar
```

At this stage, if you have followed our recommendation about using a /SERVICE/usr/lpp/PortalServer directory to perform the SMP/E steps, you should make a backup copy of the file system to another one.

For our installation we decided to copy it to /usr/lpp/PortalServer, which is the default `#WPS#` installation directory for Portal Server for z/OS.

Authorization settings

You need to set sufficient authorization to the userids with which you will use to perform the installation. In our example, TSO user `JAVA3` was given UID=0 by issuing the following TSO command:

```
ALU JAVA3 OMVS(UID(0))
```

Install PTF2 files into your portal directory

A JCL procedure installs the PTF2 files into the WebSphere Portal directory and also prepares for removing the PTF. This JCL procedure is executed with root authority.

To install PTF2 files into the WebSphere Portal directory, perform the following:

- Log on to a TSO session with root authority, UID=0).
- Execute the command:

```
oget '/usr/lpp/PortalServer/PTF/PortalServer/ejppiPTF.jcl'
'<yourhlq>.JCL(ejppiptf)' TEXT
```

- Modify the job card to meet your system requirements.
- Set the variable `BBOLIB` to the HLQ dataset qualifier for the WebSphere Application Server installation. The default value is `BBO`, however this might be different on your system.
- Replace all occurrences of the following strings:
  - `#WPS#` with the HFS location for the WebSphere Portal installation; the default is `/usr/lpp/PortalServer`

  – #DRVNUM# with 4.1.2

- Submit:

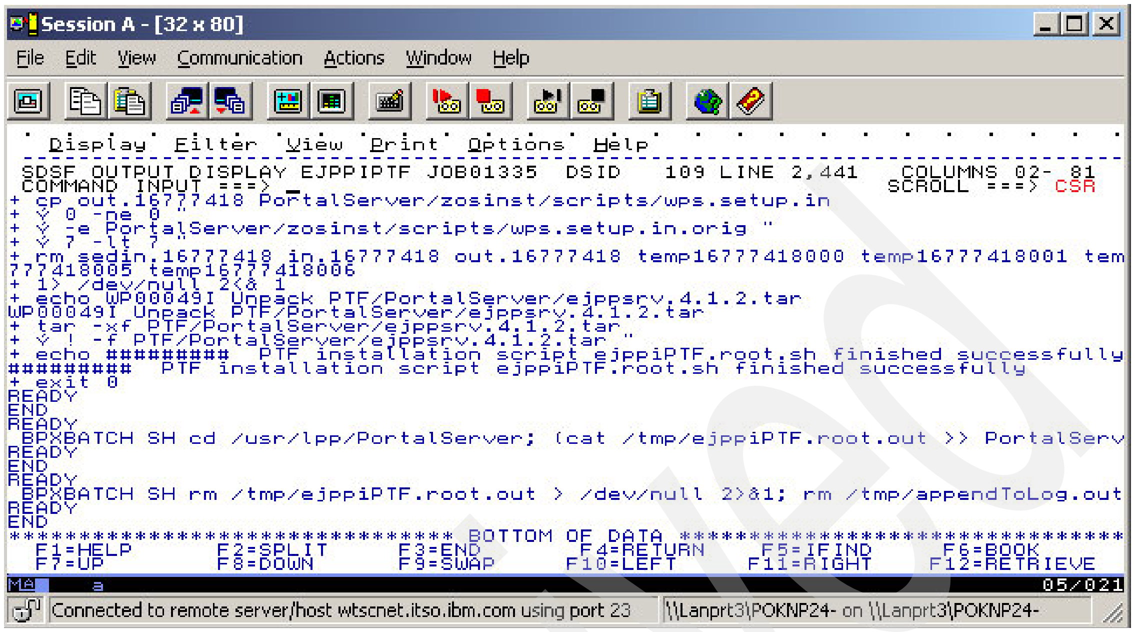
  `<yourhlq>.jobs.cntl(ejppiptf)`

This job runs the shell script `ejppiPTF.root.sh` using `BPXBATCH`. If a problem occurs, check the job log for errors, fix the problem, and restart the job.

Note: The shell script `ejppiPTF.root.sh` performs the following:

- Checks the authority of the caller.
- Loops over all files in `ejppsrv.4.1.2.tar` and saves the original files for removing the PTF. Modified files are saved to `ejppsrv.undo.4.1.2.tar`, the names of new files to `ejppnew.undo.4.1.2.txt`.
- Saves the original version of all files listed in `ejppmod.4.1.2.txt` to `ejppsrv.undo.4.1.2.tar` for removing the PTF.
- Saves the original version of all files listed in `ejppdel.4.1.2.txt` to `ejppsrv.undo.4.1.2.tar` for removing the PTF.
- Deletes all files listed in `ejppdel.4.1.2.txt` from /usr/lpp/PortalServer.
- Modifies all files as listed in `ejppmod.4.1.2.txt` in /usr/lpp/PortalServer.
- Unpacks `ejppsrv.4.1.2.tar` to /usr/lpp/PortalServer.

The highest return code from the job should be 0 and the end of that job should be similar to our display in Figure 2-4 on page 36.



```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY EJPPIPTF JOB01335  DSID   109 LINE 2,441    COLUMNS 02- 81
 COMMAND INPUT ===> _                                           SCROLL ===> CSR
+ cp out.16777418 PortalServer/zosinst/scripts/wps.setup.in
+ ÿ 0 -ne 0 "
+ ÿ -e PortalServer/zosinst/scripts/wps.setup.in.orig "
+ ÿ 7 -lt 7 "
+ rm sedin.16777418 in.16777418 out.16777418 temp16777418000 temp16777418001 tem
77418005 temp16777418006
+ 1> /dev/null 2<& 1
+ echo WP00049I Unpack PTF/PortalServer/ejppsrv.4.1.2.tar
WP00049I Unpack PTF/PortalServer/ejppsrv.4.1.2.tar
+ tar -xf PTF/PortalServer/ejppsrv.4.1.2.tar
+ ÿ ! -f PTF/PortalServer/ejppsrv.4.1.2.tar "
+ echo #########  PTF installation script ejppiPTF.root.sh finished successfully
#########  PTF installation script ejppiPTF.root.sh finished successfully
+ exit 0
READY
END
READY
 BPXBATCH SH cd /usr/lpp/PortalServer; (cat /tmp/ejppiPTF.root.out >> PortalServ
READY
END
READY
 BPXBATCH SH rm /tmp/ejppiPTF.root.out > /dev/null 2>&1; rm /tmp/appendToLog.out
READY
END
******************************** BOTTOM OF DATA ********************************
  F1=HELP        F2=SPLIT       F3=END         F4=RETURN      F5=IFIND       F6=BOOK
  F7=UP          F8=DOWN        F9=SWAP       F10=LEFT       F11=RIGHT      F12=RETRIEVE
MA     a                                                                      05/021
```

Session A - [32 x 80]
File Edit View Communication Actions Window Help
Connected to remote server/host wtscnet.itso.ibm.com using port 23
\\Lanprt3\POKNP24- on \\Lanprt3\POKNP24-

Figure 2-4 WebSphere Portal Server PTF2 files installation

2.5.3 Step three: Save the log files

Setup the proper log files (that is, `/.../Install.log`) as soon as possible, since it would be helpful to look at the log entries when a problem occurs. For this redbook project we found the following logs useful:

- /usr/lpp/PortalServer/PortalServer/Install.log
- /usr/lpp/PortalServer/PortalServer/zosinst/scripts/wps.admin.log

Note: You can create a HFS log file that is separated from the WebSphere Portal install path, /usr/lpp/PortalServer/PortalServer, because the log file can grow in size over time in a production environment.

All messages issued by the installation jobs are written to the job log and additionally, for some jobs, to a log file in the HFS. Furthermore, a combined installation log that comprises all information from the various log files is written to the file `<WPS_PATH>/Install.log`. You should save the job log of each of these jobs, and the combined installation log. These job logs are used for debugging, and are required if you contact IBM support.

2.5.4 Step four: Edit the setup file

The installation of WebSphere Portal is done by customizing and submitting several jobs for:

- WLM setup
- RACF setup
- DB2 setup
- LDAP setup
- Setup on WebSphere Application Server for z/OS and OS/390
- Portlet deployment

Generally, if a problem occurs, check the available logs, fix the problem, and restart the job. Usually the job will continue from where it previously failed. However, this might not apply to every section of the installation, in which case you must follow the instructions in those sections and also any guidelines in those sections explaining how to continue after a problem.

In some of the steps we describe how to recover when the job failed once and can't be rerun.

The file `<WPS_PATH>/zosinst/scripts/wps.setup.in` contains environment variables that need to be set prior to installation of WebSphere Portal. You can either directly edit the file <WPS_PATH>/zosinst/scripts/wps.setup.in, or copy it and edit the copy. If you do the latter, you need to supply the complete path name of your copy as a parameter to the JCL procedures for setup, and also make sure that the group CBWSPORT has read access to this directory. The advantage of copying the file <WPS_PATH>/zosinst/scripts/wps.setup.in to another location is that it allows you to migrate from one WebSphere Portal release to another without losing your changes to the file, for example, by switching or overwriting the WebSphere Portal HFS.

To make things simple, we decided to use the original file, back-up the original one, and at the end of the whole installation back-up the customized file outside of the product.

Things to remember

Agree on the naming convention for all environment variables and use the table provided in the *README: WebSphere Portal on z/OS and OS/390 Version 4.1 PTF2, Step four: Edit the setup file*, as a template. This README file can be found at:

`http://publib.boulder.ibm.com/pvc/wp/415/zos/en/readme41ptf2_install_zos.html`

- After completing the template, it should be used throughout the rest of the installation process.

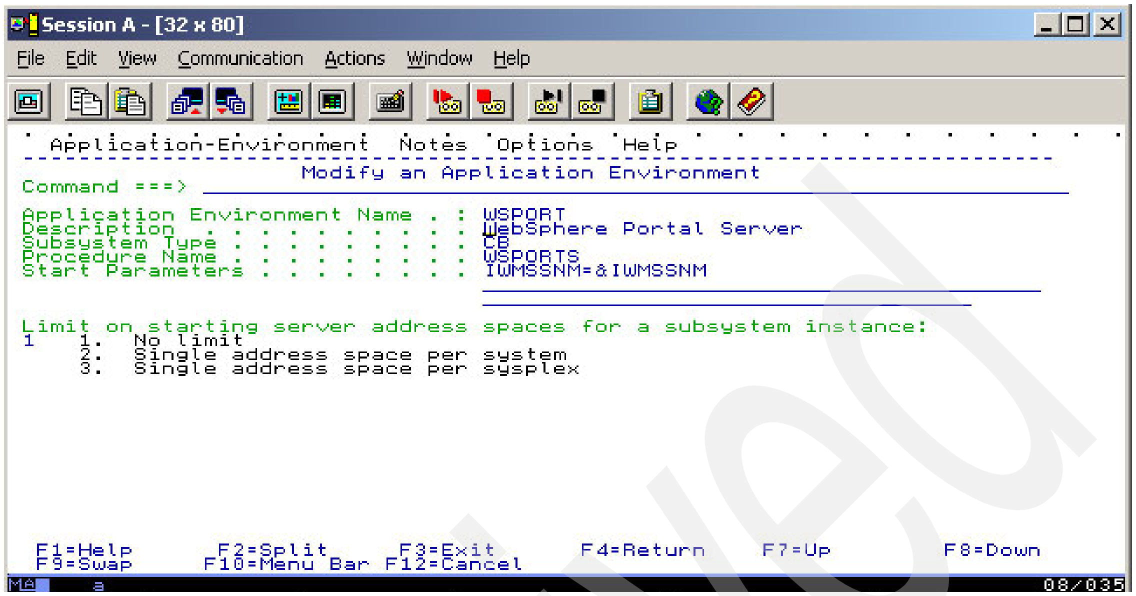
- Our experience was to take the default values from the table, that is, installation path of WebSphere Portal was /usr/lpp/PortalServer, simply because we did not have to tailor an existing environment. Refer to Table A-1 on page 325 for the environment variable settings that were used for the redbook's Portal environment.
- The installation default values for the LDAP suffix are set to '`dc=test,dc=com`'. If you want to use any value other than the default suffix then you should modify the '`initdb`' JCL under Step 8 on page 53 and replace `dc=test` to the value of your choice. For this redbook project's Portal environment we replaced `dc=test` with `dc=ibm`.

Attention: As of the writing of this book, Portal Server V4.1 supports only a two-level LDAP DN suffix. Therefore, when setting up WPS_LDAP_Server (the fully-qualified name of the LDAP server used by the WebSphere portal) and WPS_LDAP_ID, make sure your domain name uses no more than two levels.
For example: '`dc=ibm,dc=com`', instead of '`dc=itso,dc=ibm,dc=com`'

If your installation requires greater than a two-level LDAP tree name then you can use o='a.b.c',dc=x,dc=com|org|edu|net or ou='a.b.c' or dc='a.b.c'

2.5.5 Step five: Set up WLM

In the TSO environment of the target system, make sure that WLM is set up correctly using `IMWINAR0`, as shown in the following WLM definition for the server WPS_SERVERNAME in Figure 2-5 on page 39.



```
  Application-Environment  Notes  Options  Help
 ----------------------------------------------------------------------------
                    Modify an Application Environment
Command ===> ______________________________________________________________

Application Environment Name . : WSPORT
Description . . . . . . . . . .  WebSphere Portal Server
Subsystem Type . . . . . . . .   CB
Procedure Name . . . . . . . .   WSPORTS
Start Parameters . . . . . . .   IWMSSNM=&IWMSSNM
                                 ________________________________________
                                 ______________________________________

Limit on starting server address spaces for a subsystem instance:
1    1.  No limit
     2.  Single address space per system
     3.  Single address space per sysplex




 F1=Help       F2=Split      F3=Exit       F4=Return     F7=Up         F8=Down
 F9=Swap      F10=Menu Bar  F12=Cancel
MA     a                                                                08/035
```

Figure 2-5 Setup WLM Application Environment for Portal Server J2EE server

2.5.6 Step six: Set up RACF

In the TSO environment of the target system, make sure that RACF is set up correctly, as shown in Example 2-8. It shows the RACF security definition, which is used to secure WebSphere Portal Server. The user IDs created must match the user IDs specified in `wps.setup.in`. Review the RACF settings with your RACF administrator to ensure they meet your system requirements.

Example 2-8 RACF definitions for Portal J2EE server

```
/* define group cbwsport*/
/* WPS_GROUP CBWSPORT */
ADDGROUP CBWSPORT OMVS(GID(4020))
/* give access to logstream */
PERMIT #LOGSTREAM# CLASS(LOGSTRM) ID(CBWSPORT) ACCESS(UPDATE)
/* create and add wsportsu to cbwsport, configure UID to your system */
/* WPS_SRVREGIONID WSPORTSU */
ADDUSER WSPORTSU DFLTGRP(CBWSPORT) OMVS(UID(4021) HOME(/tmp) +
PROGRAM(/bin/sh)) NAME('WPS SR') PW USER(WSPORTSU) NOINTERVAL
/* create and add wsportcu to cbctl1, configure UID to your system */
/* WPS_CTLREGIONID WSPORTCU */
ADDUSER WSPORTCU DFLTGRP(CBCTL1) OMVS(UID(4022) HOME(/tmp) +
PROGRAM(/bin/sh)) NAME('WPS CR')
/* WPS_PROCNAME WSPORT */
RDEF CBIND CB.BIND.WSPORT UACC(READ)
```

```
PERMIT CB.BIND.WSPORT CLASS(CBIND) ID(CBCTL1) ACCESS(CONTROL)
RDEF CBIND CB.WSPORT UACC(READ)
RDEF SERVER CB.*.WSPORT UACC(NONE)
PERMIT CB.*.WSPORT CLASS(SERVER) ID(WSPORTSU) ACC(READ)
RDEF STARTED WSPORT.* STDATA(USER(WSPORTCU) GROUP(CBCTL1) +
TRACE(NO))
/* Server region procedure WSPORTS */
RDEF STARTED WSPORTS.* STDATA(USER(WSPORTSU) GROUP(CBWSPORT) +
TRACE(NO))
CONNECT WSPORTSU GROUP(CBCFG1)
CONNECT WSPORTCU GROUP(CBCFG1)
/* Add CBSYMSR1 and CBADMIN to group CBWSPORT */
CONNECT CBSYMSR1 GROUP(CBWSPORT) AUTH(USE)
CONNECT CBADMIN GROUP(CBWSPORT) AUTH(USE)
PERMIT CBS390 CLASS(PTKTDATA) ID(CBWSPORT) ACC(READ)
SETROPTS RACLIST(PTKTDATA) REFRESH
SETROPTS RACLIST(CBIND) GENERIC(CBIND) REFRESH
SETROPTS RACLIST(SERVER) GENERIC(SERVER) REFRESH
SETROPTS RACLIST(STARTED) GENERIC(STARTED) REFRESH
```

Important: In our case, this was a new Portal installation, so we have changed **RALT** to **RDEF** as highlighted in Example 2-8. In the README you will see:

** WPS_PROCNAME WSPORT */*
RALT STARTED WSPORT.* STDATA(USER(WSPORTCU) GROUP(CBCTL1) TRACE(NO))

/ Server region procedure WSPORTS */*
RALT STARTED WSPORTS.* STDATA(USER(WSPORTSU) GROUP(CBWSPORT) TRACE(NO))

2.5.7 Step seven: Configure LDAP server

LDAP can be configured with either RDBM or TDBM backend database systems. For better performance, we recommend that you use a TDBM backend and the TDBM setup is described here. We also recommend that you configure an LDAP server that is separate from the LDAP server used for WebSphere Application Server.

Configuring the LDAP server

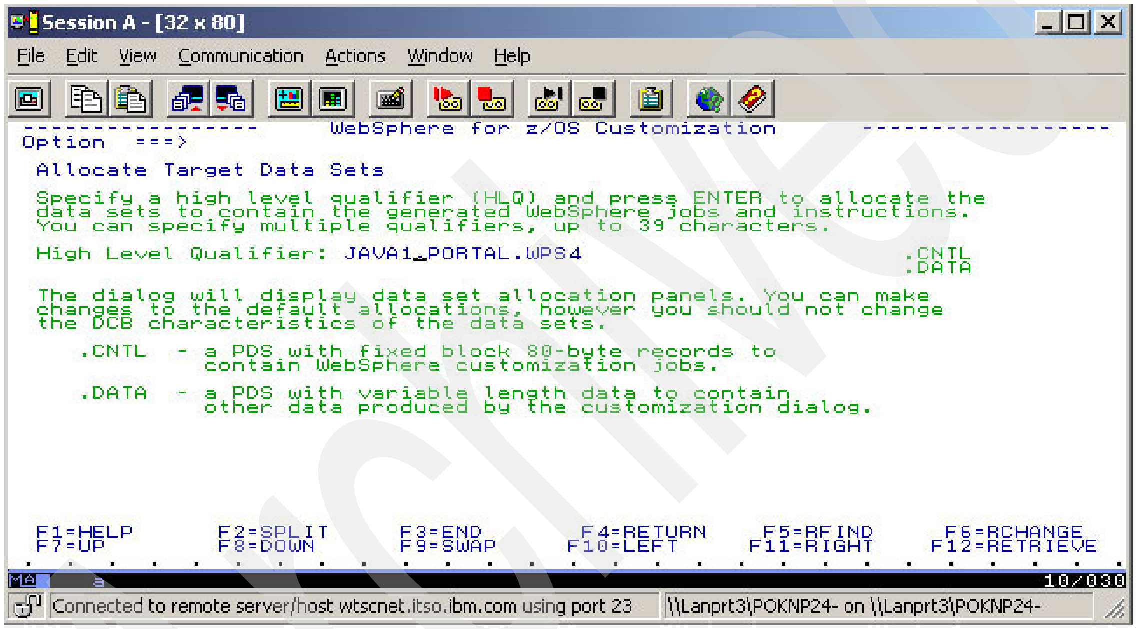
To configure the LDAP server, perform the following steps:

1. Start the Customization Dialog of WebSphere Application Server for z/OS by going to TSO and issuing command **`ex 'bbo.sbboclib(bbowstrt)'`**.

2. From the Dialog screen choose **New customization**.
3. Load the customization variables of WebSphere Application Server.
4. Allocate target data sets and name them differently to the target data sets of WebSphere Application Server. The following dataset names are generated as a sample by the configuration dialog of WebSphere Application Server, and will be different on your system:

   – TEST.PORTAL.WPS4.CNTLTEST.PORTAL.WPS4.DATA

   Here, instead of TEST as a HLQ we recommend that you use, for instance, the user name with which you are performing the Portal for z/OS installation.



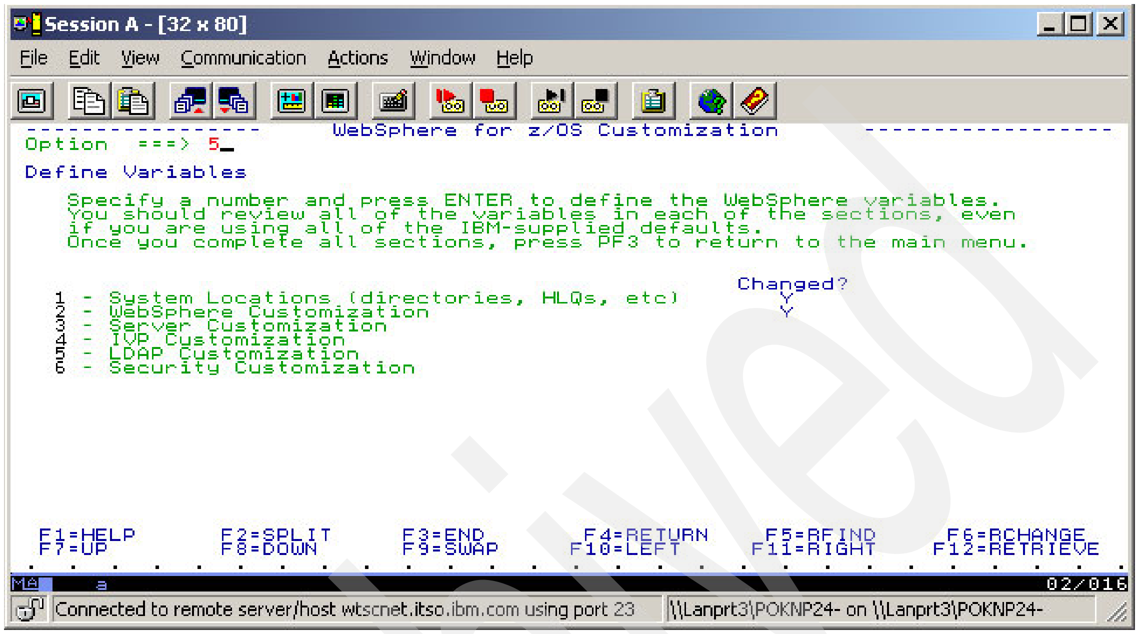
```
------------------       WebSphere for z/OS Customization      ------------------
Option ===>

 Allocate Target Data Sets

 Specify a high level qualifier (HLQ) and press ENTER to allocate the
 data sets to contain the generated WebSphere jobs and instructions.
 You can specify multiple qualifiers, up to 39 characters.

 High Level Qualifier: JAVA1.PORTAL.WPS4                                .CNTL
                                                                        .DATA

 The dialog will display data set allocation panels. You can make
 changes to the default allocations, however you should not change
 the DCB characteristics of the data sets.

    .CNTL   - a PDS with fixed block 80-byte records to
              contain WebSphere customization jobs.

    .DATA   - a PDS with variable length data to contain
              other data produced by the customization dialog.




 F1=HELP       F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
 F7=UP         F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE
                                                                          10/030
Connected to remote server/host wtscnet.itso.ibm.com using port 23
```

Figure 2-6   WebSphere Application Server installation screens - Allocate new datasets

5. To define variables, choose `LDAP Customization`. The database name, IP port, and procedure name must be different to the LDAP configuration of WebSphere Application Server. We also recommended that you use different names for the following variables, for example, we used the following:

   – Procedure name.: `BBOLDAT`
   – IP Port................: `2389`
   – DB2 STOGROUP value.....: `BBOLDSTT`
   – DB2 database name......: `BBOLDAT`
   – Authid for DB2 tables.: `BBOLDAT`
   – Database Type: `TDBM`

To do this, choose Option 5 ‘LDAP Customization’ in Define Variables as shown in Figure 2-7.

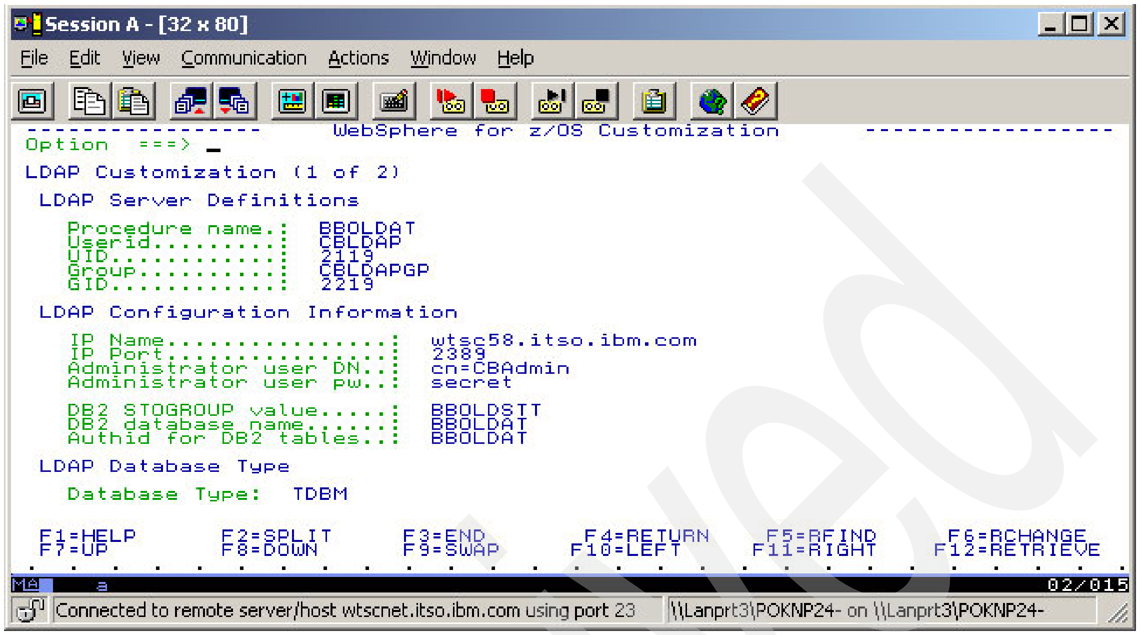


```
Session A - [32 x 80]
------------------     WebSphere for z/OS Customization      ------------------
Option  ===> 5_

Define Variables

   Specify a number and press ENTER to define the WebSphere variables.
   You should review all of the variables in each of the sections, even
   if you are using all of the IBM-supplied defaults.
   Once you complete all sections, press PF3 to return to the main menu.

                                                           Changed?
  1 - System Locations (directories, HLQs, etc)               Y
  2 - WebSphere Customization                                 Y
  3 - Server Customization
  4 - IVP Customization
  5 - LDAP Customization
  6 - Security Customization




 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA     a                                                                02/016
Connected to remote server/host wtscnet.itso.ibm.com using port 23   \\Lanprt3\POKNP24- on \\Lanprt3\POKNP24-
```

Figure 2-7 WebSphere Application Server installation screens - Define variables for new LDAP server

You will then have two different screens in which you have to customize the settings of your LDAP Server for Portal z/OS. You can use our choices displayed on Figure 2-8 on page 43 and Figure 2-9 on page 44.

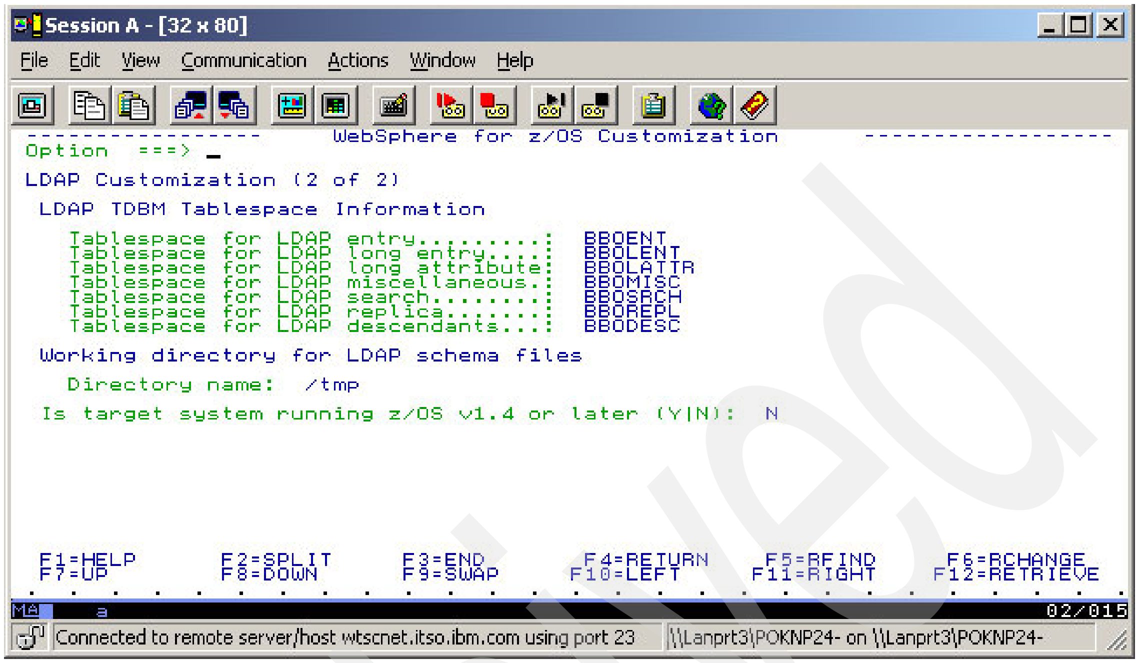
> **Note:** We are defining and preparing to install a new instance of LDAP server, not using the same LDAP that was defined for use by WebSphere Application Server. This is due to the possibility that the LDAP used by WebSphere Application Server may or may not have been defined as TDBM database type. For the redbook project environment we preferred to have separate LDAPs: BBOLDAP for WebSphere Application Server and BBOLDAT for Portal.
>
> IBM recommends that you create a separate LDAP instance for Portal server use.



```
 ------------------      WebSphere for z/OS Customization       ------------------
 Option  ===> _
 LDAP Customization (1 of 2)
  LDAP Server Definitions
    Procedure name.:   BBOLDAT
    Userid.........:   CBLDAP
    UID............:   2119
    Group..........:   CBLDAPGP
    GID............:   2219
  LDAP Configuration Information
    IP Name................:   wtsc58.itso.ibm.com
    IP Port................:   2389
    Administrator user DN..:   cn=CBAdmin
    Administrator user pw..:   secret
    DB2 STOGROUP value.....:   BBOLDSTT
    DB2 database name......:   BBOLDAT
    Authid for DB2 tables..:   BBOLDAT
  LDAP Database Type
    Database Type:   TDBM

  F1=HELP       F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
  F7=UP         F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE
MA     a                                                                    02/015
Connected to remote server/host wtscnet.itso.ibm.com using port 23   \\Lanprt3\POKNP24- on \\Lanprt3\POKNP24-
```

Figure 2-8 WebSphere Application Server installation screen 1- Define variables for new LDAP server

Screen two of defining LDAP variables for TDBM is shown in Figure 2-9 on page 44.



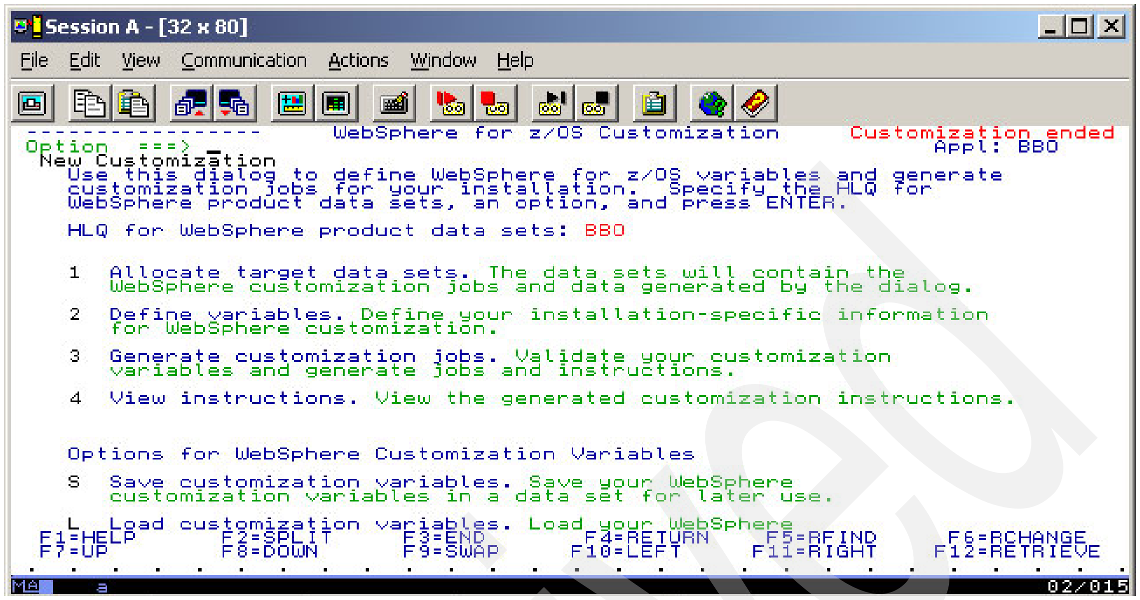
```
 -----------------        WebSphere for z/OS Customization       ------------------
 Option  ===> _
 LDAP Customization (2 of 2)
  LDAP TDBM Tablespace Information
    Tablespace for LDAP entry.........:  BBOENT
    Tablespace for LDAP long entry....:  BBOLENT
    Tablespace for LDAP long attribute:  BBOLATTR
    Tablespace for LDAP miscellaneous.:  BBOMISC
    Tablespace for LDAP search........:  BBOSRCH
    Tablespace for LDAP replica.......:  BBOREPL
    Tablespace for LDAP descendants...:  BBODESC
  Working directory for LDAP schema files
    Directory name:  /tmp
  Is target system running z/OS v1.4 or later (Y|N):  N


  F1=HELP       F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
  F7=UP         F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE
MA     a                                                                     02/015
Connected to remote server/host wtscnet.itso.ibm.com using port 23    \\Lanprt3\POKNP24- on \\Lanprt3\POKNP24-
```

Figure 2-9 WebSphere Application Server installation screen 2- Define variables for new LDAP server

6. Generate the customization jobs as shown in Figure 2-10 on page 45.
7. Save the customization variables in a dataset other than where the variables of WebSphere Application Server are kept. For example:

   ```
   <YOURHLQ>.PORTAL.WPS4.SAVECFG
   ```

8. Exit the Customization dialog.



```
Session A - [32 x 80]
File  Edit  View  Communication  Actions  Window  Help

-----------------        WebSphere for z/OS Customization      Customization ended
Option ===>                                                          Appl: BBO
 New Customization
   Use this dialog to define WebSphere for z/OS variables and generate
   customization jobs for your installation.  Specify the HLQ for
   WebSphere product data sets, an option, and press ENTER.

   HLQ for WebSphere product data sets: BBO

   1  Allocate target data sets. The data sets will contain the
      WebSphere customization jobs and data generated by the dialog.

   2  Define variables. Define your installation-specific information
      for WebSphere customization.

   3  Generate customization jobs. Validate your customization
      variables and generate jobs and instructions.

   4  View instructions. View the generated customization instructions.


   Options for WebSphere Customization Variables

   S  Save customization variables. Save your WebSphere
      customization variables in a data set for later use.

   L  Load customization variables. Load your WebSphere
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA     a                                                                 02/015
```

Figure 2-10   Generate jobs and save customization variables for LDAP server for Portal z/OS

9. Rename member `BBOLDAP` to `BBOLDAT` in the `<YOURHLQ>.PORTAL.WPS4.CNTL` dataset, and change it as shown in Example 2-9. The change is marked in **bold** type.

Example 2-9   BBOLDAT job customization

```
//BBOLDAT PROC REGSIZE=0M,
// PARMS='',
// OUTCLASS='O',
// CBCONFIG='/local/WebSphere390/CB390'
// SET LPATH='etc/ldap'
// SET RELPATH='controlinfo/envfile'
// SET LCONF='bboslapd.conf.portal'
// SET LH='/usr/lpp/ldap'
//BBOLDAT EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
// PARM=('/&PARMS >DD:SLAPDOUT 2>&1')
//STEPLIB DD DISP=SHR,DSN=SYS1.DSN.V710.SDSNLOAD
//CONFIG DD PATH='&CBCONFIG/&SYSPLEX/&LPATH/&SYSNAME..&LCONF'
//DSNAOINI DD PATH='&CBCONFIG/&SYSPLEX/&LPATH/&SYSNAME..dsnaoini'
//ENVVAR DD PATH='&LH/etc/slapd.envvars'
//SLAPDOUT DD SYSOUT=&OUTCLASS,FREE=CLOSE,SPIN=UNALLOC
//SYSOUT DD SYSOUT=&OUTCLASS,FREE=CLOSE,SPIN=UNALLOC
//SYSUDUMP DD SYSOUT=&OUTCLASS,FREE=CLOSE,SPIN=UNALLOC
//CEEDUMP DD SYSOUT=&OUTCLASS,FREE=CLOSE,SPIN=UNALLOC
```

10. Store the contents of Example 2-10 as member BBOPOLDF in the <YOURHLQ>.PORTAL.WPS4.DATA dataset using an FTP or ISPF editor. For your convenience the contents of Example 2-10 is provided in the directory <WPS_PATH>/zoinst/jcl.

BBOPOLDF member in the TEST.PORTAL.WPS4.DATA dataset:
If you have provided a LDAP suffix variable other than the dc=test default value, you will need to edit the BBOPOLDF to locate all occurrences of 'dc=test' and replace them with the variable value of your choice.

In this example you will have to customize two variables: <YOURDOMAIN> and <YOURDOMAINSUFFIX>. In our case we choose dc=ibm,dc=com instead of the default which was dc=test,dc=com.

Example 2-10 Member BBOPOLDF

```
dn: dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
o: WPS_ROOT
objectclass: top
objectclass: domain
dc: <YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
dc: <YOURDOMAIN>
description: WPS USER REGISTRY
userPassword: wpsbind
ownersource: dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
entryowner:
access-id:uid=wpsbind,cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
aclpropagate: TRUE
ownerpropagate: TRUE
aclsource: dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
aclentry:
access-id:uid=wpsbind,cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>:normal:rws
c:object:ad
aclentry: group:CN=ANYBODY:normal:rsc

dn: cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
objectclass: container
objectclass: top
cn: users

dn: cn=groups,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
objectclass: top
objectclass: container
cn: groups

dn: uid=wpsadmin,cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
objectclass: organizationalPerson
```

```
objectclass: person
objectclass: top
objectclass: inetOrgPerson
uid: wpsadmin
userpassword: wpsadmin
sn: admin
givenName: wps
cn: wps admin

dn: uid=wpsbind,cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: wpsbind
userpassword: wpsbind
sn: bind
givenName: wps
cn: wps bind

dn: cn=wpsadmins,cn=groups,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
dn: cn=wpsbind,cn=groups,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
objectclass: groupOfUniqueNames
objectclass: top
uniquemember: uid=wpsadmin,cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>
cn: wpsadmins
```

Customizing the LDAP server startup initialization file

1. Change the member BBOWTMCP in the <YOURHLQ>.PORTAL.WPS4.DATA dataset as shown in Example 2-11. The changes are marked in bold type.

Example 2-11 BBOWTMCP member - parameters for Portal LDAP Server startup

```
BBOWBOWN '<YOURHLQ>.PORTAL.WPS4.DATA(BBOWSLPT)
/local/WebSphere390/CB390/WPSPLEX/etc/ldap/WPS4.bboslapd.conf.portal 2104 2300'
BBOWBOWN '<YOURHLQ>.PORTAL.WPS4.DATA(BBOWILDF) /tmp/schema.IBM.ldif 2104 2300'
BBOWBOWN '<YOURHLQ>.PORTAL.WPS4.DATA(BBOWULDF) /tmp/schema.user.ldif 2104 2300'
BBOWBOWN '<YOURHLQ>.PORTAL.WPS4.DATA(BBOLTDBM) /tmp/bboltdbm.sh 2104 2300'
BBOWBOWN '<YOURHLQ>.PORTAL.WPS4.DATA(BBOPOLDF) /tmp/portal.ldif 2104 2300'
```

Important: You must change the LDAP configuration file name, otherwise the current file, the one used by WebSphere Application Sever for z/OS, will be overwritten! For the last line that you add, use the same UID, GID, and the same working directory; see the WebSphere Application Server Customization dialog for these (in the example shown, this is /tmp).

2. Change the member BBOLTDBM in the <YOURHLQ>.PORTAL.WPS4.DATA dataset as shown in Example 2-12. The changes are marked in bold type.

Example 2-12 TDBM backend customization file - ldapmodify statements

```
LDAPUSERID=$1
LDAPPWD=$2
ldapmodify -h 127.0.0.1 -p 2389 -D $LDAPUSERID -w $LDAPPWD -f
/tmp/schema.user.ldif
ldapmodify -h 127.0.0.1 -p 2389 -D $LDAPUSERID -w $LDAPPWD -f
/tmp/schema.IBM.ldif
ldapmodify -a -h 127.0.0.1 -p 2389 -D $LDAPUSERID -w $LDAPPWD -f
/tmp/portal.ldif
```

Important: Note that there are only five lines in Example 2-12 dataset member BBOLTDBM, such that each of the lines beginning with ldapmodify must be contained wholly within a single line.

3. Customize your LDAP server procedure. Change and add the following lines in member BBOWSLPT in the <YOURHLQ>.PORTAL.WPS4.DATA dataset:
   - Change the value of securePort to 2636
   - Remove all suffixes and add at the bottom the suffix: "dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>"
4. Run the RACF job as shown in Example 2-13.

Example 2-13 Setup RACF security for Portal z/OS LDAP server started procedure

```
//RDEF EXEC PGM=IKJEFT01,DYNAMNBR=75
//*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RDEFINE STARTED BBOLDAT.* STDATA(USER(CBLDAP) GROUP(CBLDAPGP))
SETROPTS RACLIST(STARTED) GENERIC(STARTED) REFRESH
SETROPTS RACLIST(FACILITY) GENERIC(FACILITY) REFRESH
/*
```

5. Copy member <YOURHLQ>.PORTAL.WPS4.CNTL(BBOLDAT) to USER.PROCLIB or the proclib dataset where you usually copy WebSphere Application Server jobs or LDAP jobs.

   Note: For the redbook project environment we copied members to SYS1.PROCLIB instead of USER.PROCLIB.

6. Run the jobs that reside in <YOURHLQ>.PORTAL.WPS4.CNTL, by executing the steps above; the following jobs are generated:

– `BBOTDBMC`: Creates LDAP database and table space for TDBM backend.

– `BBOLDGRT`: Grants access to the LDAP databases.

– `BBOWMCPT`: Copies the `bboslapd.conf.portal` file into WS_CONFIG_PATH/SYSPLEXNAME/etc/ladp)

– `BBOMTDBM`: Primes the LDAP database.

Job BBOTDBMC

First you must give your current userid `DB2 SYSADM authority`.

Upon completion of the job, you should see the following messages in the job log:

- `COMMIT`
- `DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION`

Job BBOLDGRT

First you must give your current userid `DB2 SYSADM authority`.

You may get return code 4 and the message:

- `THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR`

This is OK. Otherwise, you should see messages like the following in the job log:

- `GRANT DBADM ON DATABASE BBOLDAT TO CBLDAP`
- `DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION`

Job BBOWMCPT

You must run with user ID root: UID=0

Upon completion, you should see the following message for each of the four files copied in the job log:

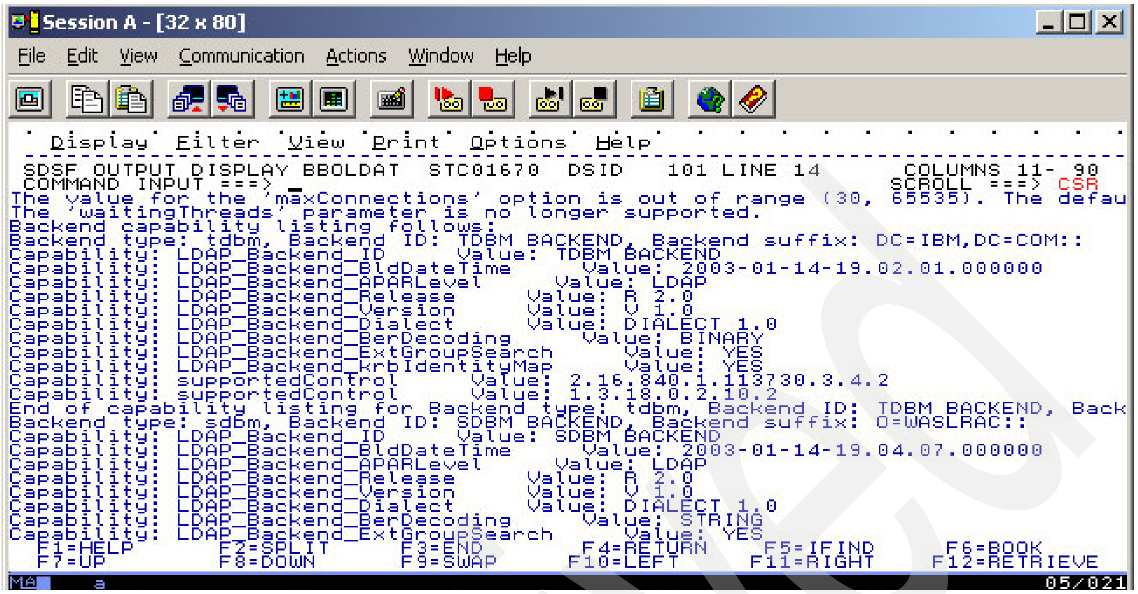
- `EXEC has completed with Return Code 0`

You can now start your LDAP server for Portal for z/OS by issuing the following TSO command: **`/S BBOLDAT,`** or equivalent for the name you chose for your environment.

Look in the SYSPRINT AND SYSLOG for message:

- `GLD0122I Slapd is ready for requests.`

You should also look for the information about what LDAP tree the TDBM backend is handling. As shown in Figure 2-11 on page 50, the TDBM backend is handling `dc=ibm,dc=com`. In your case, you should find something like dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>.


Session A - [32 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help

```
 SDSF OUTPUT DISPLAY BBOLDAT  STC01670  DSID   101 LINE 14       COLUMNS 11- 90
 COMMAND INPUT ===> _                                           SCROLL ===> CSR
The value for the 'maxConnections' option is out of range (30, 65535). The defau
The 'waitingThreads' parameter is no longer supported.
Backend capability listing follows:
Backend type: tdbm, Backend ID: TDBM BACKEND, Backend suffix: DC=IBM,DC=COM::
Capability: LDAP_Backend_ID      Value: TDBM BACKEND
Capability: LDAP_Backend_BldDateTime      Value: 2003-01-14-19.02.01.000000
Capability: LDAP_Backend_APARLevel      Value: LDAP
Capability: LDAP_Backend_Release      Value: R 2.0
Capability: LDAP_Backend_Version      Value: V 1.0
Capability: LDAP_Backend_Dialect      Value: DIALECT 1.0
Capability: LDAP_Backend_BerDecoding      Value: BINARY
Capability: LDAP_Backend_ExtGroupSearch      Value: YES
Capability: LDAP_Backend_krbIdentityMap      Value: YES
Capability: supportedControl      Value: 2.16.840.1.113730.3.4.2
Capability: supportedControl      Value: 1.3.18.0.2.10.2
End of capability listing for Backend type: tdbm, Backend ID: TDBM BACKEND, Back
Backend type: sdbm, Backend ID: SDBM BACKEND, Backend suffix: O=WASLRAC::
Capability: LDAP_Backend_ID      Value: SDBM BACKEND
Capability: LDAP_Backend_BldDateTime      Value: 2003-01-14-19.04.07.000000
Capability: LDAP_Backend_APARLevel      Value: LDAP
Capability: LDAP_Backend_Release      Value: R 2.0
Capability: LDAP_Backend_Version      Value: V 1.0
Capability: LDAP_Backend_Dialect      Value: DIALECT 1.0
Capability: LDAP_Backend_BerDecoding      Value: STRING
Capability: LDAP_Backend_ExtGroupSearch      Value: YES
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA      a                                                                 05/021
```

Figure 2-11   Check the TDBM backend suffix handled

Job BBOMTDMB

This job will initialize the LDAP database with the information you gave when customizing member `BBOPOLDF`.

Submit this job using user `CBADMIN` in your job card or any WebSphere Application Server administrator if you are not using CBADMIN for that purpose.

Tip: If you have an LDAP browser client, for example the IBM LDAP Data Management Tool called DMT, then this is a good time to login and browse the LDAP tree and check that you have added suffix `dc=ibm,dc=com` as shown in Figure 2-12 on page 51.

The IBM LDAP Data Management Tool (DMT) is shipped as part of the WebSphere Portal server distributed package on CD as part of IBM Secureway Directory Server. If you have other non-IBM LDAP browser clients, then they can be used instead.

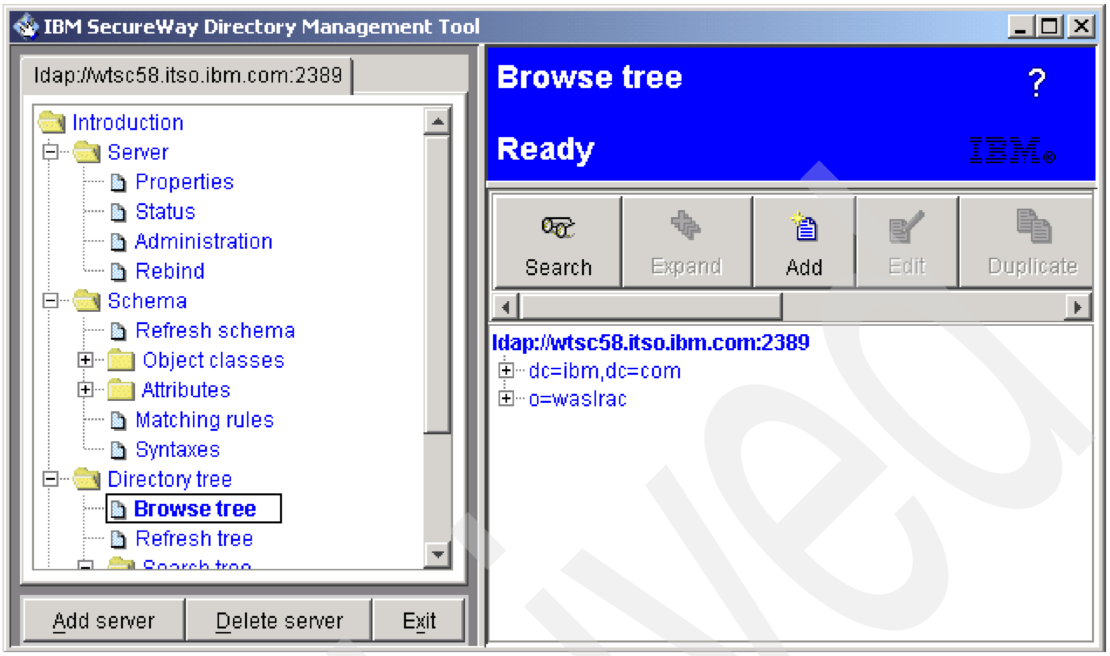


Figure 2-12 Browse LDAP tree

2.5.8 Step eight: Configure DB2 for portal

In the TSO environment of the target system, use the JCL procedure `createdb` to create the database, `inittabl` to create the database tables, and `initdb` to fill the database. The JCL procedures are located at `<WPS_PATH>/zosinst/jcl`.

The JCL procedures must be submitted with a user ID that has sufficient DB2 access rights for setting a current SQLID, usually `DB2 SYSADM`. Furthermore, the user ID must have an OMVS segment to allow for writing the WebSphere Portal installation log to HFS.

To create and install the DB2 tables follow these steps:

1. Retrieve the jobs from HFS to dataset members by executing the commands shown in Example 2-14.

Example 2-14 Retrieve portal database creation jobs

```
oget '<WPS_PATH>/zosinst/jcl/createdb.jcl' '<yourhlq>.jobs.cntl(createdb)' TEXT
oget '<WPS_PATH>/zosinst/jcl/dropdb.jcl' '<yourhlq>.jobs.cntl(dropdb)' TEXT
oget '<WPS_PATH>/zosinst/jcl/initdb.jcl' '<yourhlq>.jobs.cntl(initdb)' TEXT
oget '<WPS_PATH>/zosinst/jcl/inittabl.jcl' '<yourhlq>.jobs.cntl(inittabl)' TEXT
oget '<WPS_PATH>/zosinst/jcl/wpsPrMsg.clist' '<yourhlq>.jobs.cntl(wpsprmsg)'
TEXT
```

2. Modify the job cards of the JCL procedures to meet your system requirements.
3. Replace all occurrences of the following strings in the JCL procedures:

   The DB2 parameters to be replaced in the JCL are not obvious to people without strong DB2 background. To consult a database administrator (DBA) is the best way to find these parameter values. If a DBA is not reachable, you can look up the job BBOCBGRT of WebSphere Customization JCL to find the correct values of #DB2PREFX#, #DB2SUB#, and #PLAN#.

   – Replace #DB2PRFX# with the high level qualifier of the DB2 data sets on your system. The DSN= parameter on DBRMLIB DD card provides the #DB2PREFX#
   – Replace #DB2SUB# with the name of the DB2 Group Attach name for datasharing environments or the DB2 subsystem name for non-datasharing environments. The DSN SYSTEM statement provides #DB2SUB#
   – Replace #PLAN# with the name of the plan for the DSNTIAD program on your system (for example, DSNTIA71 for DB2 release 7.1).
     You can check the RUN PROGRAM(DSNTIAD) PLAN statement, that provides #PLAN# or issue the following SQL statements in DB2 SPUFI or QMF™:

     ```
     SELECT NAME, CREATOR from sysibm.sysplan where NAME LIKE 'DSNTI%'
     ```

   – Replace #WPS_SRVREGIONID# with the name of server region user ID of the WebSphere Portal. This user ID must have sufficient DB2 access rights for creating databases, table spaces, and tables.
   – Replace #WPS# with the HFS location for the WebSphere Portal installation; the default is /usr/lpp/PortalServer.
   – Replace #YOURHLQ# with the high level qualifier of the JCL procedure dataset of WebSphere Portal on your system.

   Make sure the RUNLIB, where the application plan `#PLAN#` is defined, is specified in the RUN PROGRAM statement:

   ```
   RUN PROGRAM(DSNTIAD) PLAN(DSNTIA71) -
      LIB('DB2V710A.RUNLIB.LOAD')
   ```

   You can find the correct RUNLIB from JCL, BBOCBGRT, also. If a proper RUNLIB is not included, you may get a `DSNTIAD plan not found in SYS1.LOAD` error message when executing these three jobs.

   In our example `#WPS#` matched `/usr/lpp/PortalServer`. In your installation it has to match <WAS_ROOT>.

4. Copy member INITDB to another member, for example INITDC and edit it. Locate the two references to dc=test,dc=com and modify them according to the choices you made for the DN suffix during LDAP configuration. The two references to dc=test,dc=com are two inserts in the Portal for z/OS database that prime the admin user and group information to be used by Portal as shown in Example 2-15.

Example 2-15 Copy and modify the INITDB job

```
(...)
INSERT INTO USER_DESC ( OID, NAME, TYPE, CREATED, MODIFIED )
VALUES ( 10, 'uid=wpsadmin,cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>', 1,
            1022882400000, 1022882400000 );
INSERT INTO USER_DESC ( OID, NAME, TYPE, CREATED, MODIFIED )
VALUES ( 11, 'cn=wpsadmins,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>', 2,
            1022882400000, 1022882400000 );
(...)
```

5. Submit '<yourhlq>.jobs.cntl(createdb)'

Note: We ran into a JCL error with missing UNIT=xxx parameter, you may need to provide a UNIT=value to successfully run the job

6. Verify you receive the message ######### Installation job CREATEDC finished successfully in the job log or in the installation log located at <WPS_PATH>/Install.log of WebSphere Portal. If a problem occurs, check the job log or installation log for errors, fix the problem, and restart the job.

Note: You can ignore the following error message in DROP DATABASE WPS41 statements:

```
WCPDLL RC=8 from drop data base WCPDB

DSNT408I SQLCODE = -204, ERROR: WPS41 IS AN UNDEFINED NAME
```

7. Submit '<yourhlq>.jobs.cntl(inittabl)'
8. Verify you receive the message ######### Installation job INITTABL finished successfully in the job log or in the installation log <WPS_PATH>/Install.log of WebSphere Portal. If a problem occurs, check the job log or installation log for errors, fix the problem, and resubmit the job.
9. Submit '<yourhlq>.jobs.cntl(initdb)'

Note: If you are not using the default LDAP suffix of dc=test,dc=com, make sure you edit the `initdb.jcl`. Locate all instances of ‘`dc=test,dc=com`’ and provide the LDAP suffix of your choice, before running the initdb job.

10. Verify you receive the message `####### Installation job INITDB finished successfully` in the job log or in the installation log `<WPS_PATH>/Install.log` of WebSphere Portal.

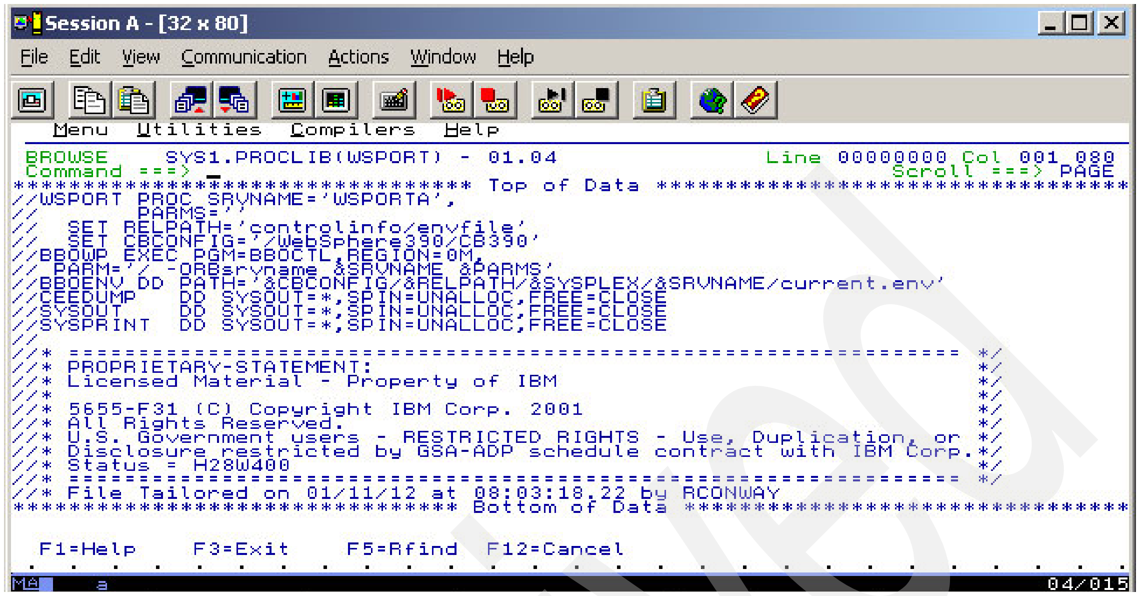
Note: You can ignore the following warning message in DELETE FROM... statements:

```
DSNT404I SQLCODE = 100, NOT FOUND: ROW NOT FOUND FOR FETCH, UPDATE, OR
DELETE,....
```

11. Create JCL procedures for the control and server regions:

    To do this, follow the instructions in the “Steps for creating JCL procedures for the control and server regions” in Chapter 8 of *WebSphere Application Server for z/OS and OS/390 Version 4.01: Assembling Java(TM) 2 Platform Enterprise Edition (J2EE) Applications,* SA22-7836.

    Samples created using those guidelines are provided in directory <WPS_PATH>/zosinst/jcl. You can also copy JCL procedures you already created for the J2EE IVP of WebSphere Application Server or any JCL procedure that successfully starts a J2EE application server, and tailor it for that new server as shown in Figure 2-13 on page 55.


Session A - [32 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

```
BROWSE    SYS1.PROCLIB(WSPORT) - 01.04              Line 00000000 Col 001 080
Command ===>                                                  Scroll ===> PAGE
****************************** Top of Data ***********************************
//WSPORT PROC SRVNAME='WSPORTA',
//          PARMS=''
//   SET RELPATH='controlinfo/envfile'
//   SET CBCONFIG='/WebSphere390/CB390'
//BBOWP EXEC PGM=BBOCTL,REGION=0M,
// PARM='/ -ORBsrvname &SRVNAME &PARMS'
//BBOENV DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&SRVNAME/current.env'
//CEEDUMP    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT     DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//
//* ================================================================ */
//* PROPRIETARY-STATEMENT:                                           */
//* Licensed Material - Property of IBM                              */
//*                                                                  */
//* 5655-F31 (C) Copyright IBM Corp. 2001                            */
//* All Rights Reserved.                                             */
//* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
//* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
//* Status = H28W400                                                 */
//* ================================================================ */
//* File Tailored on 01/11/12 at 08:03:18.22 by RCONWAY
****************************** Bottom of Data ********************************

  F1=Help    F3=Exit    F5=Rfind  F12=Cancel
MA      a                                                               04/015
```

Figure 2-13 WSPORT JCL for Portal z/OS control region started task

2.5.9 Step nine: Install Portal on WebSphere Application Server

On the z/OS and OS/390 operating systems, WebSphere Portal is installed as a J2EE server running on WebSphere Application Server. Three JCL procedures are used to configure the portal on WebSphere Application Server for z/OS and OS/390. The first and last JCL procedures are executed with root authority. The second JCL procedure contains the parameters `USER=` and `PASSWORD=` in the job card, and therefore runs under administrative authority for WebSphere Application Server, for example, userID CBADMIN.

To install WebSphere Portal, perform the following:

1. Log on to a TSO session with root authority, UID=0. Execute the commands shown in Example 2-16:

Example 2-16 Retrieve basic Portal for z/OS installation JCL

```
oget '<WPS_PATH>/zosinst/jcl/wpsPre.jcl' '<yourhlq>.jobs.cntl(wpspre)' TEXT
oget '<WPS_PATH>/zosinst/jcl/wpsAdmin.jcl' '<yourhlq>.jobs.cntl(wpsadmin)' TEXT
oget '<WPS_PATH>/zosinst/jcl/wpsPost.jcl' '<yourhlq>.jobs.cntl(wpspost)' TEXT
oget '<WPS_PATH>/zosinst/jcl/WPAConf.jcl' '<yourhlq>.jobs.cntl(wpaconf)' TEXT
oget '<WPS_PATH>/zosinst/jcl/WPAIns.jcl' '<yourhlq>.jobs.cntl(wpains)' TEXT
```

2. Modify the job cards to meet your system requirements. Set the variable `BBOLIB` with the HLQ dataset qualifier for the WebSphere Application Server

installation. The default value is `BBO`, however, this might be different on your system.

3. Replace all occurrences of the following strings:
   - Replace `#WPS#` with the HFS location for the WebSphere Portal installation; the default is /usr/lpp/PortalServer.
   - Replace `#INPUTFILE#` with the HFS location of the WebSphere Portal input file. Do one of the following:

     If you copied WPS_PATH/zosinst/scripts/wps.setup.in in an earlier step and have edited the copy, you need to specify the complete path of your copy here.

     If you have directly edited WPS_PATH/zosinst/scripts/wps.setup.in, remove the `#INPUTFILE#` variable.
   - Replace `#WASADMIN#` with the user ID of a WebSphere Application Server administrator, for example, `CBADMIN`. Do this only for the `wpsAdmin.jcl` and `WPAConf.jcl` jobs.
   - Replace `#WASADMPW#` with the password of the user ID added for `#WASADMIN#`.
   - `#JOBCOUNT#`: Remove this string to make sure that all available jobs are processed. Do this only for `WPAConf.jcl`.
4. Submit: `<yourhlq>.jobs.cntl(wpspre)`
5. Submit: `<yourhlq>.jobs.cntl(wpsadmin)`
6. Submit: `<yourhlq>.jobs.cntl(wpspost)`

Tip: The ‘wpsadmin’ job may take up to 15 minutes to run. Therefore, ensure the `TIME=value` parameter on the JOB card is set accordingly, otherwise the job will time-out with `ABEND522`.

Update Portal to use the defined LDAP suffix

You now need to update files: `um.properties`, `VaultService.properties` and `authtable_wcp.xml` of Portal Server, to configure the LDAP suffix you defined in “Step seven: Configure LDAP server” on page 40.

The files listed above are located in <WPS_PATH>/libapp/config directory and are stored in ASCII, so you may need to FTP them to a workstation and edit them there. Modify the files to make the DN suffix match the DN suffix of your choice, as shown in our example (Example 2-17 on page 57).

Example 2-17 Update um.properties file

```
(...)
# default templates for user/group lookup if no DN is given

# WES:
#user.template=uid={0},cn=users,ou=IBM,dc=ibm,dc=com
user.template=uid={0},cn=users,dc=ibm,dc=com
user.template.attribute=uid
group.template=cn={0},cn=groups,dc=ibm,dc=com
group.template.attribute=cn


# lookup parameters for looking up groups for users

group.lookup.context=cn=groups,dc=ibm,dc=com
group.lookup.filter=(&(objectclass=groupOfUniqueNames)(uniqueMember={0}))
group.lookup.attributes=dn,cn
group.member.attribute=uniqueMember
user.lookup.context=cn=users,dc=ibm,dc=com

# To support multiple LDAP implementations all keys starting with
"java.naming."
# are automatically inserted into the LDAP context.

java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
#java.naming.factory.initial=com.ibm.jndi.LDAPCtxFactory
#java.naming.factory.url.pkgs=com.ibm.jndi
java.naming.provider.url=ldap://wtsc58.itso.ibm.com:2389
java.naming.security.principal=uid=wpsbind,cn=users,dc=ibm,dc=com
java.naming.security.credentials=wpsbind

(...)
```

2.5.10 Step ten: Set up multiple location names (optional)

This step is optional and we did not set up multiple location names when writing the redbook.

2.5.11 Step eleven: Start WebSphere Portal

The base installation is complete after the successful configuration of WebSphere Portal on WebSphere Application Server for z/OS or OS/390, performed in steps nine and ten. Next, you need to perform a verification of the installation.

To start WebSphere Portal for verification of the base portal functionality, use the following command from the z/OS or OS/390 system:

```
S WPS_PROCNAME.WPS_SERVERNAMEA,
```

For example, we used:

```
S WSPORT.WSPORTA
```

During the first startup of Portal Server it will perform the naming registration for the portlets and display console messages about starting and completing the registration. Wait until the naming registration has completed, and view the empty home page from a browser at:

```
http://WPS_HOST:WPS_PORT/wps/WPS_PUBLIC_HOME
```

For example, we viewed the home page at:

```
http://wtsc58.itso.ibm.com:8082/wps/portal
```

You should receive a Web page with a message stating that the portal contains no page groups. This is expected at this stage and you can still login to the Portal Server by clicking on the key icon in the banner of the portal's Web page.

Log in to WebSphere Portal with user `wpsadmin` password `wpsadmin` and verify that the login was successful.

Note: If all previous steps were completed successfully, you should receive the messages from the Portal server console and log file, `open for business`. If not, use the error code and message from the console and log file to determine the cause of the error.

2.5.12 Step twelve: Deploy the base portlets

Deploying the base portlets requires two steps that submit a job at each step.

The first step requires that Portal is running, and a job is used to execute an internal deployment tool that deploys the base portlets into the portal configuration, and create description files for the second step.

During the second step of deployment a script called `WPAConfig` reads the description files created in Step 1, calls the *390fy* tool to transform the portlet war files to ear files, and calls SM API to define the portlets as J2EE applications on WebSphere Application Server.

Deploy Step 1

To execute the first step of deploying the base portlets on WebSphere Portal, perform the following steps. First, make sure your Portal server is still up and running from the previous step.

1. Log on to a TSO session with a user ID that has root authority, UID=0.
2. Submit job: `<yourhlq>.jobs.cntl(wpains)`

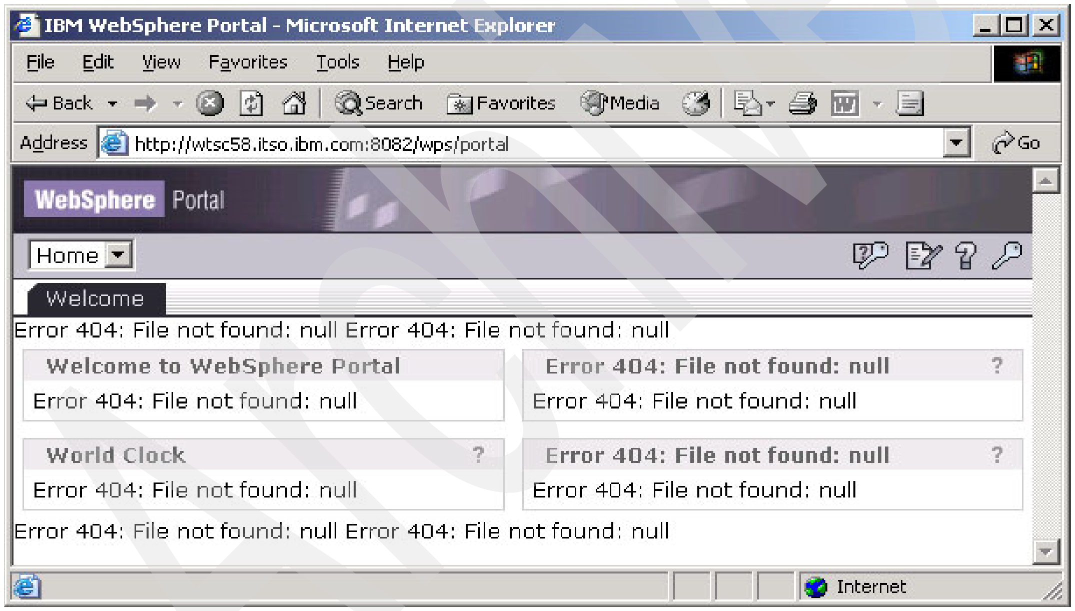
   On completion verify the success of the job by viewing the portal welcome page again from a browser.

   ```
   http://WPS_HOST:WPS_PORT/wps/WPS_PUBLIC_HOME
   ```

   For example, we viewed the page at:

   ```
   http://wtsc58.itso.ibm.com:8082/wps/portal
   ```

3. You should see the frames of four portlets with message `Error 404` as shown in Figure 2-14.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://wtsc58.itso.ibm.com:8082/wps/portal
Go
WebSphere Portal
Home
Welcome
Internet

Error 404: File not found: null Error 404: File not found: null

Welcome to WebSphere Portal
Error 404: File not found: null

Error 404: File not found: null
Error 404: File not found: null

World Clock
Error 404: File not found: null

Error 404: File not found: null
Error 404: File not found: null

Error 404: File not found: null Error 404: File not found: null

Figure 2-14 Base portlets deployment (Step 1)

Deploy step 2

To execute the second step to deploy the base portlets on WebSphere Portal, perform the following:

4. From the z/OS or OS/390 system, stop WebSphere Portal via the command:

   ```
   P WPS_PROCNAME.WPS_SERVERNAMEA
   ```

For example we used: **`PWSPORT.WSPORTA`**

Before submitting the next job, `WPACONF`, check your jobcard to see if you have a TIME parameter. Either set to a generous value or leave it at the default value of 1440, because this job will take several minutes to run to completion.

5. Submit job: `<yourhlq>.jobs.cntl(wpaconf)`
6. Check the job log for successful completion. If a problem occurred, do a search on “Verify the deployment” in the Portal InfoCenter and follow the instructions there.
7. Restart WebSphere Portal for verification of portlet deployment by using the following command on the z/OS or OS/390 system:

   `S WPS_PROCNAME.WPS_SERVERNAMEA`

   For example, we used: **`S WSPORT.WSPORTA`**

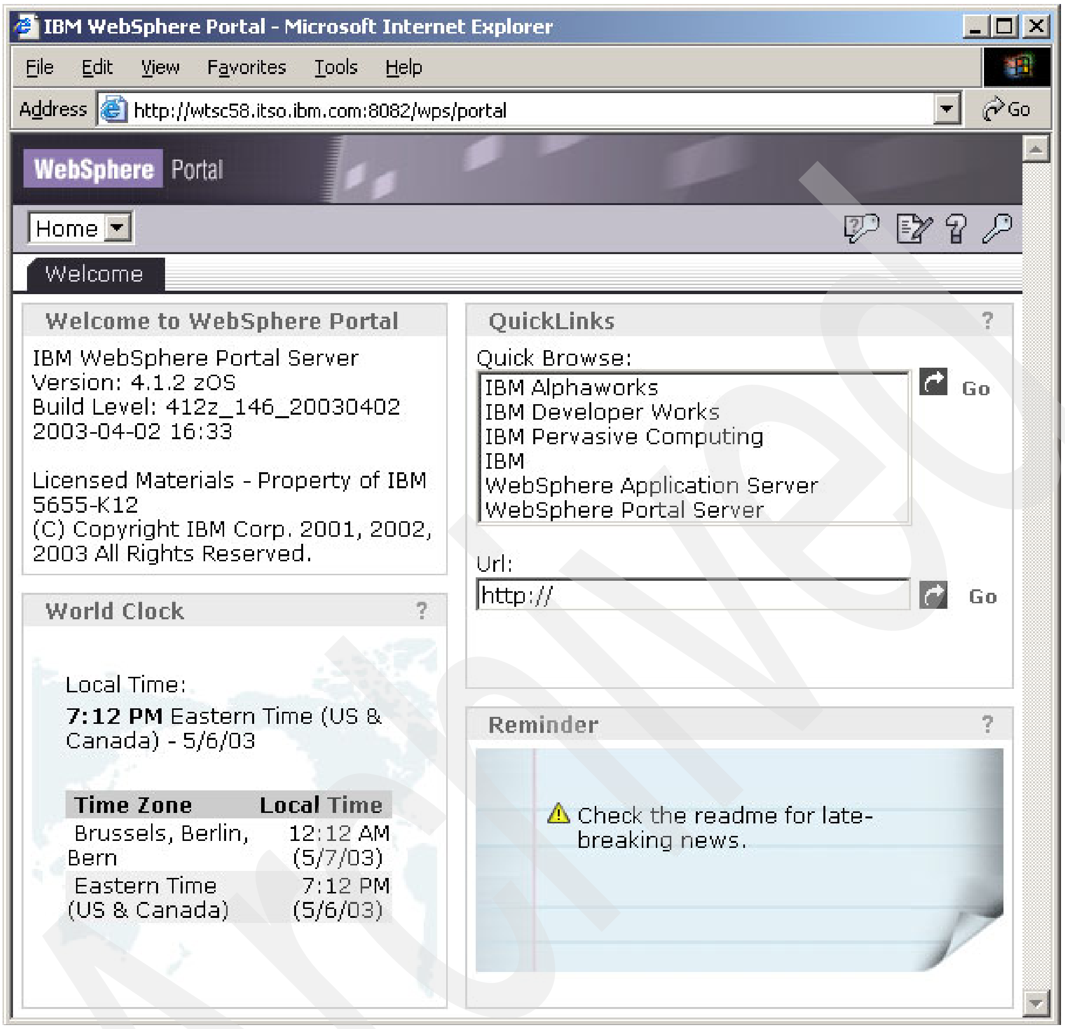
   This startup performs naming registration for the base portlets and displays console messages about starting and completing the registration.
8. Wait until the naming registration has completed, then again use a browser to view the portal home page at:

   `http://WPS_HOST:WPS_PORT/wps/WPS_PUBLIC_HOME`

   For example we used:

   `http://wtsc58.itso.ibm.com:8082/wps/portal`

You should now be able to see the complete portal welcome page as shown in Figure 2-15 on page 61, following successful deployment of the base portlets.


IBM WebSphere Portal - Microsoft Internet Explorer

Address http://wtsc58.itso.ibm.com:8082/wps/portal

Welcome to WebSphere Portal

IBM WebSphere Portal Server
Version: 4.1.2 zOS
Build Level: 412z_146_20030402
2003-04-02 16:33

Licensed Materials - Property of IBM
5655-K12
(C) Copyright IBM Corp. 2001, 2002, 2003 All Rights Reserved.

World Clock

Local Time:
7:12 PM Eastern Time (US & Canada) - 5/6/03

Time Zone	Local Time
Brussels, Berlin, Bern	12:12 AM (5/7/03)
Eastern Time (US & Canada)	7:12 PM (5/6/03)

QuickLinks

Quick Browse:
IBM Alphaworks
IBM Developer Works
IBM Pervasive Computing
IBM
WebSphere Application Server
WebSphere Portal Server

Url: http://

Reminder

Check the readme for late-breaking news.

Figure 2-15 Base portlets deployment (Step 2) completed

That completes post-installation of the Portal Server.

Portal migration from PTF2 to PTF3

During the post-processing of this redbook, PTF3 became available for WebSphere Portal. Here we give you a summary of what needs to be done regarding PTF3. The migration from PTF2 to PTF3 has been done in the Sysplex configuration as described in 5.1, “Configuring WebSphere Portal in a Sysplex” on page 148. Therefore, here we talk about "shared HFS" and multiple server instances.

The following description is based on the steps described in the PTF3 Readme “Migrating to WebSphere Portal for z/OS and OS/390 Version 4.1 PTF 3”. Where no action was taken for a step, “No action” is stated next to the step.

- Step one: Read about package content.
- Step two: Unpack PTF3 files.

  The SMP/E deliverable was installed on the shared HFS and the **tar** command was executed on one of systems in the sysplex:

  ```
  tar -xvf ejppsrv.4.1.2.412z_150_20030716.tar
  ```

- Step three: Save the log files.
- Step four: Install PTF 3 files.

  The JCL procedure `ejppiPTF.jcl` was copied to the z/OS data set, modified, and the job submitted.

- Step five: Migrate the DB2 database (only if migrating from PTF1).

  Our test migration is from PTF2.

- Step six: Migrate WebSphere Portal on WebSphere Application Server.

  The JCL procedures `ejppmre.jcl`, `ejppredp.jcl`, `ejppupdp.jcl`, `WPAConf.jcl`, and `wpsPre.jcl` were copied to the appropriate z/OS data set and the job files modified.

  All portal instances were stopped (using the command **stop WPS_PROCNAME.$WPS_SERVERNAMEA**, and so on).

  The jobs `wpspre`, `ejppredp`, and `ejppmre` were submitted.

  Note: If manual modifications have been made in the property files (such as, the port number in `ConfigService.properties`), these modifications have to be done again in the new files. Note that to check if modifications have been done, compare the saved files in `…/PTF/PortalServer/ejppsrv.undo.<driver number>.tar` with the new property files after migration.

  The portal instance was restarted:

  ```
  S WPS_PROCNAME.$WPS_SERVERNAMEA,srvname='$WPS_SERVERNAMEA'
  ```

- Step seven: Migrate portlets on WebSphere Portal.

  The job `ejppupdp` was submitted to migrate the portlets.

  The running portal instance was stopped:

  ```
  stop WPS_PROCNAME.$WPS_SERVERNAMEA
  ```

  The job `wpaconf` was submitted.

All portal instances were restarted:

```
s WPS_PROCNAME.$WPS_SERVERNAMEA,srvname='$WPS_SERVERNAMEA'
```

2.6 WebSphere Transcoding Publisher PTF2

Here we discuss the WebSphere Transcoding Publisher post SMP/E installation. WebSphere Transcoding Publisher PTF2 is a full product replacement for PTF1 or a new installation. Therefore, the post installation is quite straightforward consisting of two main steps.

Step 1: Update property files

1. Log on to a TSO session with root authority, UID=0.
2. Copy the JCL procedure `EJPTPSET` from UNIX System Services to your authorized JCL library, `<YOURHLQ>.JCL` for instance. To do this, execute the following command from the command prompt of the ISPF Command Shell:
   – **`oget '<WTP_PATH>/JCL/EJPTPSET' '<YOURHLQ>.JCL(EJPTPSET)'`**
3. Modify the job card to meet your system requirements.
4. Set the variable `BBOLIB` with the HLQ dataset qualifier for the WebSphere Application Server installation, the default is `BBO`.
5. Replace all occurrences of the following strings:
   – Replace `#WTP#` with the HFS location for the WebSphere Transcoding Publisher installation; the default is `<WTP_ROOT>`, and in our case we used /usr/lpp/PortalServer/IBMTrans.
   – Replace `#INPUTFILE#` with the HFS location of the WebSphere Portal input file by doing one of the following:
      - If you made a copy of <WPS_PATH>/zosinst/scripts/wps.setup.in, during 2.5.4, “Step four: Edit the setup file” on page 37 of the WebSphere Portal installation, and have edited that copy, specify the complete path of your copy here.
      - If you directly edited <WPS_PATH>/zosinst/scripts/wps.setup.in, remove the `#INPUTFILE#` variable.
6. Submit the job `EJPTPSET`.

Step 2: Register plug-ins with Transcoding Publisher

1. Copy the JCL procedure from UNIX System Services to your authorized JCL library. To do this, execute the following command from the command prompt of the ISPF Command Shell:
   – **`oget '<WTP_ROOT>/JCL/EJPREGPL' '<YOURHLQ>.JCL(EJPREGPL)'`**

2. Modify the job card to meet your system requirements.
3. Set the variable `BBOLIB` with the HLQ dataset qualifier for the WebSphere Application Server installation. The default for installation is BBO.
4. Replace all occurrences of the following strings:
   - Replace `#WTP#` with the HFS location for the WebSphere Transcoding Publisher installation; the default is <WTP_ROOT>.
   - Replace `-f #INPUTFILE#` with the HFS location of the WebSphere Portal input file wps.setup.in. The *<-f>* parameter is required to select the user-defined file. Do one of the following:
     - If you made a copy of <WPS_PATH>/zosinst/scripts/wps.setup.in, during 2.5.4, "Step four: Edit the setup file" on page 37 of the WebSphere Portal installation, and have edited that copy, specify the complete path of your copy here.
     - If you directly edited <WPS_PATH>/zosinst/scripts/wps.setup.in, remove the `#INPUTFILE#` variable.
5. Submit the job `EJPREGPL`.

That completes the WebSphere Transcoding Publisher installation.

2.7 WebSphere Portal Content Publishing PTF2

Here we discuss the WebSphere Portal Content Publishing post SMP/E installation. At this point you should have successfully completed the installation of Portal Server and Transcoding Publisher and the next steps are to complete post installation of WebSphere Portal Content Publishing runtime (WPCP).

WPCP PTF2 gets installed on top of WPCP PTF1 so that PTF1 is a prerequisite even on a new installation. To do this, perform the following steps:

"Step 1: Read about package content" on page 65
"Step 2: Edit the wcp.setup.in setup file" on page 65
"Step 3: Install JCL procedure fixes" on page 66
"Step 4: Create DB2 tables" on page 66
"Step 5: Configure WebSphere Portal Content Publishing" on page 67
"Step 6: Install PTF2" on page 69
"Step 7: Verify the installation" on page 69

Important: Be very careful not to confuse PTF1 and PTF2 files during the installation steps. Read carefully through the description of each step, as some of the steps refer to files delivered with PTF2 located in /usr/lpp/PortalServer/PTF/wcp, and other steps refer to files for PTF1 in /usr/lpp/PortalServer/wcp.

Step 1: Read about package content

The installation file `ejpwcpub.tar` has already been installed by SMP/E. This file contains the following:

- `jcl` directory containing JCL procedures to create and fill the DB2 database tables and run the installation scripts.
- `ear` directory containing the files `PznRuntime.ear` and `WCPDemo_Runtime.ear`.
- `scripts` directory containing REXX and shell scripts to set up WebSphere Portal Content Publishing on WebSphere Application Server for z/OS or OS/390.
- `lib` directory containing JAR files for WebSphere Portal Content Publishing.
- `pznConfig` directory containing configuration files for WebSphere Portal Content Publishing.

These directories are located at `<WCP_PATH>` after the installation file is unpacked.

The PTF2 installation file `ejpwcpub.DRVNUM.tar` has already been installed by SMP/E. This file contains the following:

- `jcl` directory containing JCL procedures to replace some JCL files in the base installation and to install PTF2.
- `ear` directory containing fixes for the base version of `WCPDemo_Runtime.ear`.
- `scripts` directory containing shell scripts that replace some scripts in the base installation and scripts that install PTF2.
- `lib` directory containing replacements for files `personalization.jar` and `prCommon.jar`.
- `pznConfig` directory contains a replacement for `publishAdapters.properties`.

These directories are located installed at /usr/lpp/PortalServer/PTF/wcp after the installation file is unpacked.

Step 2: Edit the `wcp.setup.in` setup file

The setup file is located in WebSphere Portal Content Publishing PTF2 extracted PTF tree directory. By default this directory is /usr/lpp/PortalServer/PTF. In this example, the file to edit will be `/usr/lpp/PortalServer/PTF/wcp/wcp.setup.in`. You can see the values we used in `wcp.setup.in` in Table B-1 on page 329.

Step 3: Install JCL procedure fixes

To install fixes for JCL procedures and scripts prior to the base installation, perform the following steps:

1. Import the JCL procedure `ejpc2fx` into the JCL library to repair files for WebSphere Portal Content Publishing. To do this, execute the following command from UNIX System Services:
   – **`oget '/usr/lpp/PortalServer/PTF/wcp/jcl/ejpc2fx' '<yourhlq>.JCL(ejpc2fx)' TEXT`**
2. Select the job `ejpc2fx`.
3. Modify the job card to meet your system requirements.
4. Replace all occurrences of the following strings:
   – Replace `#WCP2#` with the HFS path of the PTF2 installation of WebSphere Portal Content Publishing. The default for this is /usr/lpp/PortalServer/PTF/wcp.
   – Replace `#INPUTFILE1#` with the fully-qualified path and name of the input setup file `wps.setup.in` of WebSphere Portal. The default location for this is usually /usr/lpp/PortalServer/PortalServer/zosinst/scripts/wps.setup.in.
   – Replace `#INPUTFILE2#` with the HFS path and file name of the input setup file `wcp.setup.in` of WebSphere Portal Content Publishing within the directory /usr/lpp/PortalServer/PTF/wcp/scripts.
   – Replace `#WEBSPHERE#` with the HLQ dataset qualifier for the WebSphere Application Server installation, default `BBO`.
5. Run the JCL procedure `ejpc2fx`.

Step 4: Create DB2 tables

In the TSO environment of the target system, use job `WCPDDL` to create and load the database table `WCPTS`. To do this, get the job from the UNIX System Services to your authorized JCL library and then submit the job by performing the following:

1. **`oget '<WCP_PATH>/jcl/WCPDDL' '<yourhlq>.JCL(WCPDDL)' TEXT`**
2. Modify the job card, DB2 name, and joblib of the JCL procedure to meet your system requirements.
3. Submit: `'<yourhlq>.JCL(WCPDDL)'`
4. Verify the successful completion by checking for the successful commit message in the last output line.

Note: You may receive an error message:

```
WCPDDL RC=8 from drop data base WCPDB

DSNT408I SQLCODE= -204 error WCPDS is an undefined name
```

This error is due to the fact that the table is to be created in this job step, and it does not exist until the job has successfully completed execution, so the message can be ignored.

Step 5: Configure WebSphere Portal Content Publishing

On the z/OS and OS/390 operating systems, WebSphere Portal Content Publishing is installed and configured as a server on WebSphere Application Server. Three JCL procedures are used to install and configure WebSphere Portal Content Publishing on WebSphere Application Server for z/OS or OS/390. To perform the configuration, follow these steps:

1. Log on to a TSO session with root authority, UID=0.
2. Copy the JCL procedure from UNIX System Services to your authorized JCL library. To do this, execute the following commands from the UNIX System Services command prompt:
   – **`oget '<WCP_PATH>/jcl/EJPCPPRE' '<yourhlq>.JCL(EJPCPPRE)' TEXT`**
   – **`oget '<WCP_PATH>/jcl/EJPCPADM' '<yourhlq>.JCL(EJPCPADM)' TEXT`**
   – **`oget '<WCP_PATH>/jcl/EJPCPPOS' '<yourhlq>.JCL(EJPCPPOS)' TEXT`**
3. Select the job `EJPCPPRE`.
4. Modify the job card to meet your system requirements.
5. Set the variable `BBOLIB` with the HLQ dataset qualifier for the WebSphere Application Server installation; by default this is `BBO`.
6. Replace all occurrences of the following strings:
   – Replace `#WCP#` with the location for the WebSphere Portal Content Publishing installation; the default is `<WCP_ROOT>`.
   – Replace `#INPUTFILE1#` with the fully-qualified path and name of the input setup file `wps.setup.in` of WebSphere Portal. This parameter is optional and defaults to: /usr/lpp/PortalServer/PortalServer/zosinst/scripts/wps.setup.in.
   – Replace `#INPUTFILE2#` with the fully-qualified path and name of the input setup file `wcp.setup.in` of WebSphere Portal Content Publishing. This parameter is optional and defaults to `WCP_PATH/scripts/wcp.setup.in`. Unlike previous steps the `#INPUTFILE2#` parameter here is mandatory.
7. Submit the job and check the return codes for successful completion.

8. From the z/OS or OS/390 console, stop WebSphere Portal using the command:

   `/P WPS_PROCNAME.WPS_SERVERNAMEA`

   For example, in our installation we used:

   `/P WSPORT.WSPORTA`

9. Select the job `EJPCPADM`.
10. Modify the job card to meet your system requirements.
11. Set the variable `BBOLIB` with the HLQ dataset qualifier for the WebSphere Application Server installation; the default is `BBO`.
12. Replace all occurrences of the following strings:
    – Replace `#WCP#` with the location for the WebSphere Portal Content Publishing installation; the default is `<WCP_PATH>`.
    – Replace `#INPUTFILE1#` with the fully-qualified path and name of the input setup file `wps.setup.in` of WebSphere Portal. This parameter is optional and defaults to: /usr/lpp/PortalServer/PortalServer/zosinst/scripts/wps.setup.in.
    – Replace `#INPUTFILE2#` with the fully-qualified path and name of the input setup file *wcp.setup.in* of WebSphere Portal Content Publishing. This parameter is optional and defaults to <WCP_PATH>/scripts/wcp.setup.in.
    – Replace `#WASADMIN#` with the user ID of a WebSphere Application Server administrator, for example, `CBADMIN`.
    – Replace `#WASADMPW#` with the password of that user ID.
13. Submit the job and check the return codes for successful completion.
14. Next select job `EJPCPPOS`.
15. Modify the job card to meet your system requirements.
16. Set the variable `BBOLIB` with the HLQ dataset qualifier for the WebSphere Application Server installation; the default is `BBO`.
17. Replace all occurrences of the following strings:
    – Replace `#WCP#` with the location for the WebSphere Portal Content Publishing installation; the default is `<WCP_PATH>`.
    – Replace `#INPUTFILE1#` with the fully-qualified path and name of the input setup file `wps.setup.in` of WebSphere Portal. This parameter is optional and defaults to: /usr/lpp/PortalServer/PortalServer/zosinst/scripts/wps.setup.in.
    – Replace `#INPUTFILE2#` with the fully-qualified path and name of the input setup file `wcp.setup.in` of WebSphere Portal Content Publishing. This parameter is optional and defaults to <WCP_PATH>/scripts/wcp.setup.in.

18. Submit the job and check the return codes for successful completion.

Step 6: Install PTF2

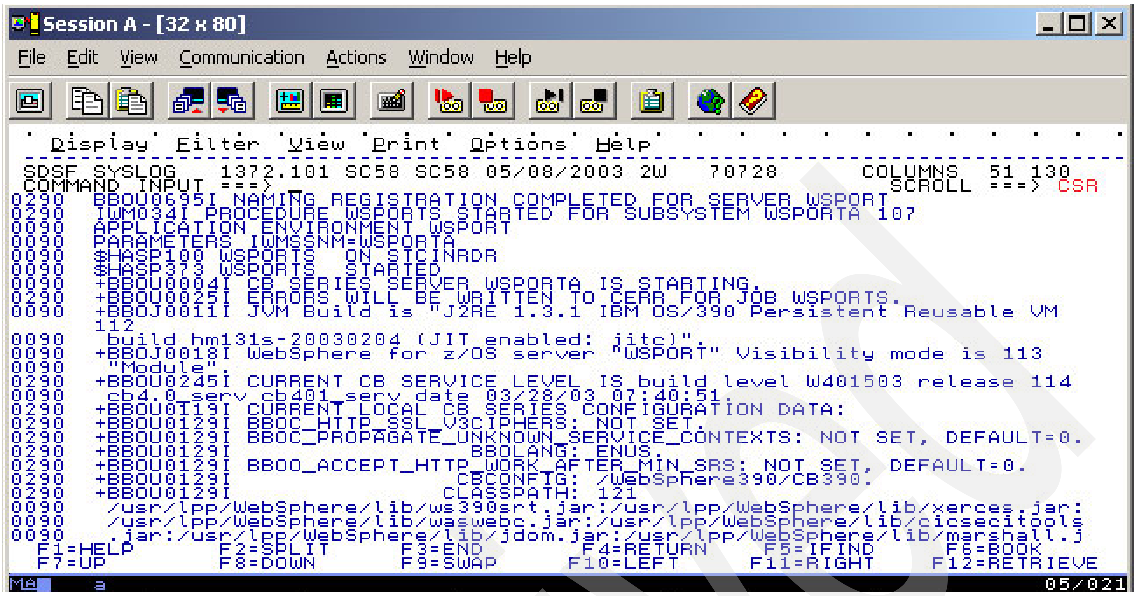
To install the PTF2 files for WebSphere Portal Content Publishing, perform the following:

1. Import the JCL procedure `ejpc2in` into the JCL library to install the PTF2 files. To do this, execute the following command from UNIX System Services:
   – **`oget '/usr/lpp/PortalServer/PTF/wcp/jcl/ejpc2in' '<yourhlq>.JCL(ejpc2in)' TEXT`**
2. Select the job `ejpc2in`.
3. Modify the job card to meet your system requirements.
4. Replace all occurrences of the following strings:
   – Replace `#WCP2#` with the HFS path of the WebSphere Portal Content Publishing installation; the default for this is usually /usr/lpp/PortalServer/PTF/wcp.
   – Replace `#INPUTFILE1#` with the HFS path and file name of the input setup file of WebSphere Portal; the default for this is usually /usr/lpp/PortalServer/PortalServer/zosinst/scripts/wps.setup.in.
   – Replace `#INPUTFILE2#` with the HFS path and file name of the input setup file of WebSphere Portal Content Publishing; the default for this is usually WCP_PATH/scripts/wcp.setup.in.
   – Replace `#WEBSPHERE#` with the HLQ dataset qualifier for the WebSphere Application Server installation; the default is `BBO`.
5. Run the JCL procedure `ejpc2in` to launch `wcp2install.sh`.

Step 7: Verify the installation

The base WPCP installation is now complete after the successful configuration of WebSphere Portal Content Publishing on WebSphere Application Server for z/OS or OS/390, and the installation of PTF2. Next, you need to verify the installation.

Start WebSphere Portal Server and wait until naming registration completes by looking in `SDSF LOG` for message `BBOU0695I` as shown in Figure 2-16 on page 70.



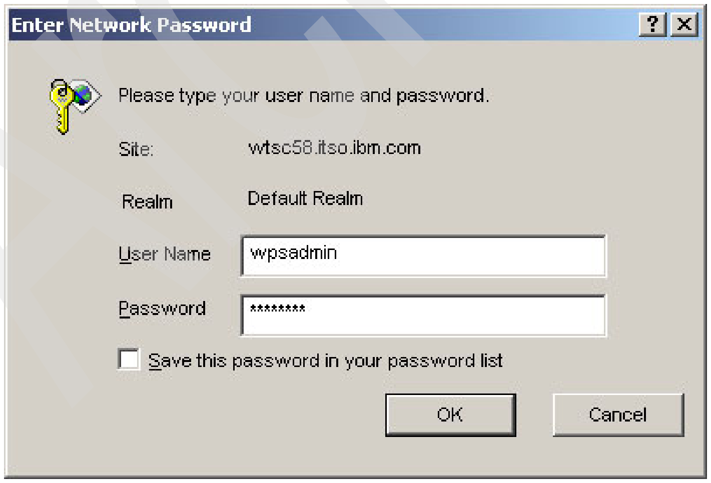
```
 Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG    1372.101 SC58 SC58 05/08/2003 2W    70728        COLUMNS  51 130
 COMMAND INPUT ===>                                                SCROLL ===> CSR
0290   BBOU0695I NAMING REGISTRATION COMPLETED FOR SERVER WSPORT
0290   IWM034I PROCEDURE WSPORTS STARTED FOR SUBSYSTEM WSPORTA 107
0090   APPLICATION ENVIRONMENT WSPORT
0090   PARAMETERS IWMSSNM=WSPORTA
0090   $HASP100 WSPORTS  ON STCINRDR
0090   $HASP373 WSPORTS  STARTED
0090   +BBOU0004I CB SERIES SERVER WSPORTA IS STARTING.
0290   +BBOU0025I ERRORS WILL BE WRITTEN TO CERR FOR JOB WSPORTS.
0090   +BBOJ0011I JVM Build is "J2RE 1.3.1 IBM OS/390 Persistent Reusable VM
       112
0090    build hm131s-20030204 (JIT enabled: jitc)".
0090   +BBOJ0018I WebSphere for z/OS server "WSPORT" Visibility mode is 113
0090    "Module".
0290   +BBOU0245I CURRENT CB SERVICE LEVEL IS build level W401503 release 114
0090    cb4.0_serv cb401_serv date 03/28/03 07:40:51.
0290   +BBOU0119I CURRENT LOCAL CB SERIES CONFIGURATION DATA:
0290   +BBOU0129I BBOC_HTTP_SSL_V3CIPHERS: NOT SET.
0290   +BBOU0129I BBOC_PROPAGATE_UNKNOWN_SERVICE_CONTEXTS: NOT SET, DEFAULT=0.
0290   +BBOU0129I                        BBOLANG: ENUS.
0290   +BBOU0129I BBOO_ACCEPT_HTTP_WORK_AFTER_MIN_SRS: NOT SET, DEFAULT=0.
0290   +BBOU0129I                       CBCONFIG: /WebSphere390/CB390.
0290   +BBOU0129I                      CLASSPATH: 121
0090    /usr/lpp/WebSphere/lib/ws390srt.jar:/usr/lpp/WebSphere/lib/xerces.jar:
0090    /usr/lpp/WebSphere/lib/waswebc.jar:/usr/lpp/WebSphere/lib/cicsecitools
0090    .jar:/usr/lpp/WebSphere/lib/jdom.jar:/usr/lpp/WebSphere/lib/marshall.j
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 2-16 WebSphere Portal Content Publishing naming registration completed

Open the WebSphere Portal Content Publishing administrative console for Personalization by using a browser to connect to this URL:

```
http://WPS_HOST:WPS_PORT/wps/PersAdmin/adminframe.jsp
```

You should receive a login challenge as shown in Figure 2-17.



Enter Network Password

Please type your user name and password.

Site:	wtsc58.itso.ibm.com
Realm	Default Realm
User Name	wpsadmin
Password	********

☐ Save this password in your password list

Figure 2-17 Personalization challenge/response based authentication

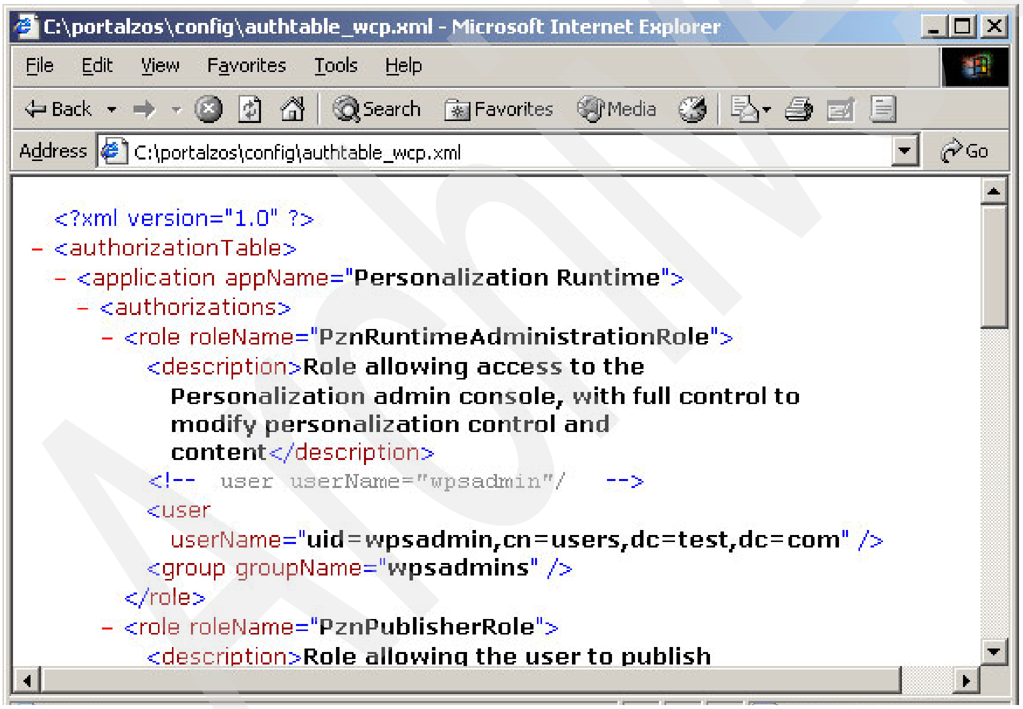
This is a prompt to login as a user authorized to the role:

`PznRuntimeAdministrationRole`

The roles of a J2EE application installed on the Portal Server are located in directory `/usr/lpp/PortalServer/PortalServer/libapp/config`, and the main configuration file is `authtablelist.xml`. This file points to a specific J2EE security file depending on the J2EE application `auth_table.xml` for Portal Server, and `authtable_wcp.xml` for all other applications.

If you examine `authtable_wcp.xml` as shown in Figure 2-18, you will notice that `PznRuntimeAdministrationRole` role is given to the `WPSADMINS` group and to `uid=wpsadmin,cn=users,dc=test,dc=com`. This may not be what you want at this stage for your environment, because you would probably want to change the hardcoded user to another, as was defined in Example 2-10 on page 46:

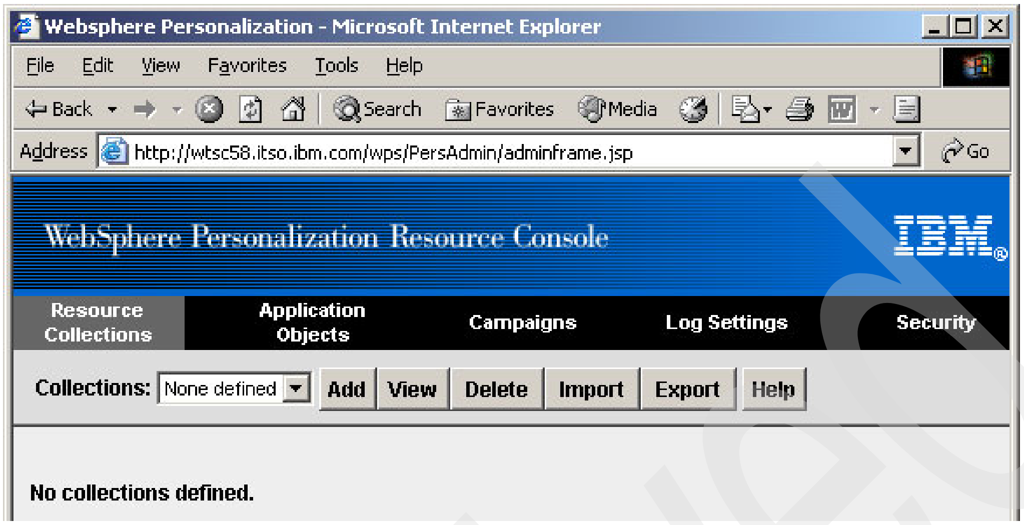
`uid=wpsadmin,cn=users,dc=<YOURDOMAIN>,dc=<YOURDOMAINSUFFIX>`


C:\portalzos\config\authtable_wcp.xml - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Address
C:\portalzos\config\authtable_wcp.xml
Go

```
  <?xml version="1.0" ?>
- <authorizationTable>
  - <application appName="Personalization Runtime">
    - <authorizations>
      - <role roleName="PznRuntimeAdministrationRole">
          <description>Role allowing access to the
            Personalization admin console, with full control to
            modify personalization control and
            content</description>
          <!--  user  userName="wpsadmin"/    -->
          <user
            userName="uid=wpsadmin,cn=users,dc=test,dc=com" />
          <group groupName="wpsadmins" />
        </role>
      - <role roleName="PznPublisherRole">
          <description>Role allowing the user to publish
```

Figure 2-18   authtable_wcp.xml file

At this stage in the installation the `wpsadmin` user with password `wpsadmin` belongs to the `WPSADMINS` LDAP group and can be used to complete the login to WPCP, after which you should receive the page shown in Figure 2-19 on page 72.


Websphere Personalization - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://wtsc58.itso.ibm.com/wps/PersAdmin/adminframe.jsp Go
WebSphere Personalization Resource Console
IBM
Resource Collections
Application Objects
Campaigns
Log Settings
Security
Collections: None defined Add View Delete Import Export Help

No collections defined.

Figure 2-19 WebSphere Personalization "logged in" screen

Note: Congratulations! You have successfully completed the configuration and installation of WebSphere Portal Server run-time environment on z/OS and OS/390. For configuration and installation of the WebSphere Portal Server *Development environment*, see Chapter 3, "Development environment installation".

2.8 Configuring the HTTP plug-in

Here we discuss configuring the WebSphere Application Server HTTP plug-in. To provide a high-end HTTP Server to our Portal Server installation we also used IBM HTTP Server (IHS) for z/OS. We installed and configured the *HTTP plug-in* for the IBM HTTP Server on z/OS which is functionally equivalent to the "distributed" plug-in.

The HTTP plug-in for IHS provides session affinity for requests flowing from a browser via the HTTP Server to Portal Server. Session affinity is required for Portal users on z/OS, when multiple server regions are configured for WLM of the portal. Therefore, unless you implement Session Persistency in the Application Server, user's requests need to always return to the same server instance, same control region, and same server region.

The HTTP plug-in running inside IBM HTTP Server for z/OS is the only solution that provides this unique feature of session affinity.

2.8.1 Edit HTTP Server httpd.conf file

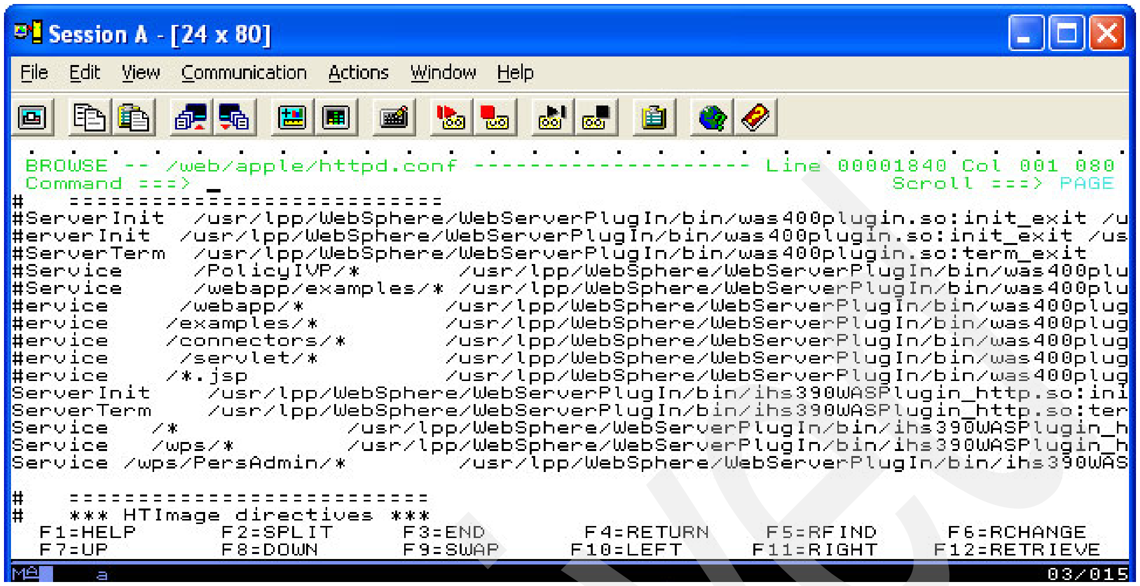
To use the HTTP plug-in with the IBM HTTP Server for z/OS, instead of the IIOP-based plug-in, you need to configure the `httpd.conf` file of the HTTP Server as follows:

1. Locate the `httpd.conf` file.
2. Remove or comment out any existing WebSphere Application Server related plug-in configuration in the file.
3. Update it according to Example 2-18.

Example 2-18 Update IBM HTTP Server configuration file httpd.conf

```
ServerInit
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:init_exit
/web/apple/plugin-cfg.xml
ServerTerm
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:term_exit
Service /*
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:service_exit
Service /wps/*
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:service_exit
Service /wps/PersAdmin/*
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:service_exit
```

Note that the statements: `ServerInit`, `ServerTerm` and `Service`, have to be written on separate lines, with parameters separated by blanks, as shown in Figure 2-20 on page 74.



```
 BROWSE -- /web/apple/httpd.conf -------------------- Line 00001840 Col 001 080
 Command ===> _                                                Scroll ===> PAGE
#    ==========================
#ServerInit  /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:init_exit /u
#erverInit  /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:init_exit /us
#ServerTerm  /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:term_exit
#Service     /PolicyIVP/*       /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plu
#Service     /webapp/examples/* /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plu
#ervice     /webapp/*          /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plug
#ervice    /examples/*         /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plug
#ervice    /connectors/*       /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plug
#ervice     /servlet/*         /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plug
#ervice    /*.jsp              /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plug
ServerInit     /usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:ini
ServerTerm     /usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:ter
Service    /*              /usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_h
Service    /wps/*          /usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_h
Service /wps/PersAdmin/*        /usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WAS

#    ==========================
#    *** HTImage directives ***
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA     a                                                                   03/015
```

Figure 2-20   Update IBM HTTP Server configuration file *httpd.con*

The `ServerInit` parameter points to a `plugin-cfg.xml` file that you need to create and tailor as explained in the next section.

2.8.2 Create a plugin-cfg.xml file

Create a new file `plugin-cfg.xml` and edit it as shown in Example 2-19.

Example 2-19   plugin-cfg.xml

```
<?xml version="1.0"?>
<Config>
    <Log LogLevel="Error" Name="/web/apple/logs/plugin.trace"/>
    <VirtualHostGroup Name="default_host">
        <VirtualHost Name="localhost:80"/>
        <VirtualHost Name="wtsc58.itso.ibm.com:80"/>
        <VirtualHost Name="wtsc58.itso.ibm.com:443"/>
</VirtualHostGroup>
    <ServerGroup Name="WTSC58/WSPORT">
   <Server Name="WSPORT">
      <Transport Hostname="wtsc58.itso.ibm.com" Port="8082" Protocol="http"/>
   </Server>
    </ServerGroup>
  <UriGroup Name="portal">
   <Uri Name="/*"/>
</UriGroup>  .
```

```
  <Route ServerGroup="WTSC58/WSPORT" UriGroup="portal"
    VirtualHostGroup="default_host"/>
</Config>
```

The parameters used in this file are as follows:

- `<VirtualHost Name="localhost:80"/>` is used by Portal to identify local connections.
- `<VirtualHost Name="wtsc58.itso.ibm.com:80"/>` is tailored to match the HTTP Server DNS name, and the port value based on the `httpd.conf` file port parameter.
- `<Transport Hostname="wtsc58.itso.ibm.com" Port="8082" Protocol="http"/>` is used by the HTTP plug-in to connect to the HTTP listeners running in the WebSphere Server Instances in the Control Regions.

2.8.3 Modify WebSphere Web container settings

Next modify the Portal J2EE Web container to accept requests coming from IBM HTTP Server for z/OS.

- Locate the `webcontainer.conf` file of the Portal J2EE server. It should be located in <WAS_ROOT>/controlinfo/envfile/yoursysplexname/portalname/.
- Edit the `webcontainer.conf` file and locate the `host.default_host.alias` variable.
- Add the HTTP Server hostname and port number from which the Portal Server will process requests, as shown in Example 2-20.

Example 2-20 Update Portal J2EE server webcontainer.conf file

```
host.default_host.alias=wtsc58.itso.ibm.com:8082, wtsc58.itso.ibm.com:80,
wtsc58.itso.ibm.com:443
```

2.8.4 Update Portal redirection mechanism

Portal Server uses a URL redirect after login authentication. After adding IHS and the plug-in, you will be able to connect through the IBM HTTP Server port, default 80, but after login the Portal will redirect the browser to the configured Control Region HTTP port, in our case port 8082. To change this behavior so that all authenticated requests also flow via the HTTP Server port you need to perform the following:

- Locate and edit `ConfigService.properties` file in the Portal Server installation. It is located in:
  <WPS_PATH>/PortalServer/libapp/config/services/

- This file is in ASCII, so you may need to transfer it to a workstation and edit it there.

  Update `ConfigService.properties` as shown in Example 2-21.

Example 2-21 ConfigService.properties

```
(...)
host.name = wtsc58.itso.ibm.com
host.port.http = 80
host.port.https =
(...)
```

The `host.name` parameter has to match the HTTP Server DNS name, also the name in `webcontainer.conf` and `plugin-cfg.xml`.

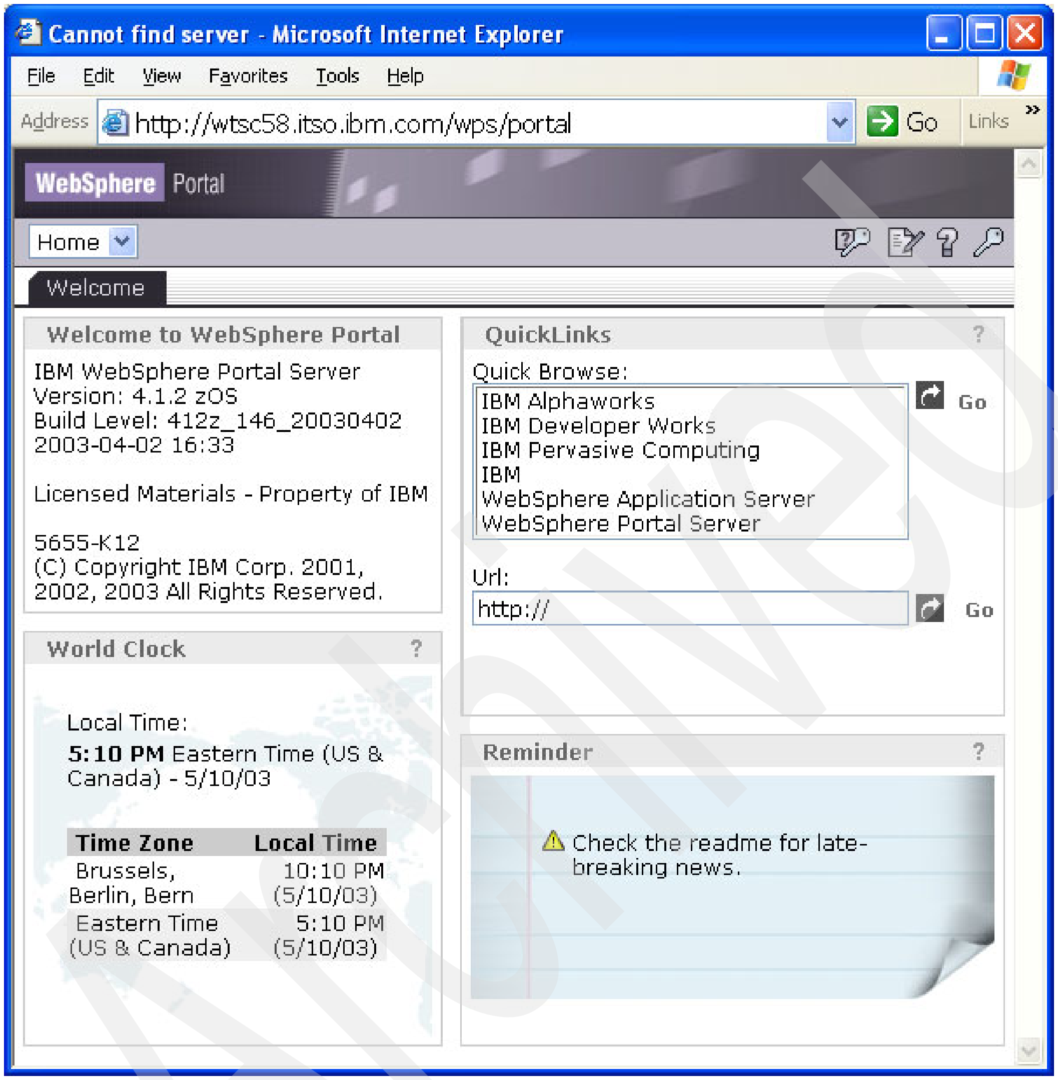
The `host.port.http` parameter has to match the one defined in `webcontainer.conf` and `plugin-cfg.xml` file.

2.8.5 Test the login redirection

To test the change to redirect after logging in:

- Stop Portal Server, for example by issuing command: **`/P WSPORTA`**
- Stop the IBM HTTP Server
- Start the IBM HTTP Server
- Start Portal Server, for example by issuing command: **`/S WSPORT.WSPORTA`**

Connect to Portal via the IBM HTTP Server using the configured port, for example in our case port 80, as shown in Figure 2-21 on page 77:


Cannot find server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://wtsc58.itso.ibm.com/wps/portal
Go
Links
WebSphere Portal
Home
Welcome

Welcome to WebSphere Portal

IBM WebSphere Portal Server
Version: 4.1.2 zOS
Build Level: 412z_146_20030402
2003-04-02 16:33

Licensed Materials - Property of IBM

5655-K12
(C) Copyright IBM Corp. 2001, 2002, 2003 All Rights Reserved.

QuickLinks

Quick Browse:
IBM Alphaworks
IBM Developer Works
IBM Pervasive Computing
IBM
WebSphere Application Server
WebSphere Portal Server

Url:
http://

World Clock

Local Time:
5:10 PM Eastern Time (US & Canada) - 5/10/03

Time Zone	Local Time
Brussels, Berlin, Bern	10:10 PM (5/10/03)
Eastern Time (US & Canada)	5:10 PM (5/10/03)

Reminder

Check the readme for late-breaking news.

Figure 2-21   Connect to Portal from HTTP Server

Next login to Portal. This time after logging in the Portal will issue a redirect using the same port that you initially connected with, on the IBM HTTP Server. Therefore after logging in, the same port should show in the URL on the browser.

You have completed the HTTP plugin setup for Portal z/OS!

2.9 Installation worksheet

Table 2-3 summarizes all the installation steps which are listed in the README file. You can use it as a worksheet with our “to-do’s” and comments, and add your own.

Table 2-3 Installation worksheet to use as a guide for your own installation

Step number and name	To-dos	Variables used	Validation	Comments
1. Read package content				There are three main components of this installation, step (4-7) to complete the LDAP Server and it's TDBM, step (5,6,8-11) to complete the Portal Server, and step (12) to deploy the base portlets. Use a LDAP browser to validate the LDAP Server is working can save lots of troubleshooting times later.
	Review Portal V4.1.2 Readme file			
	Verify that WebSphere Application Server V4.01 is up			
	Install Unicode code support			Re-IPL required to activate the Unicode support
2. Unpack WebSphere Portal files				
2.1 base product	Complete SMP/E installation			Make sure the HFS file size is at least 600 cyls. Also the /tmp size is large enough for all the log files.
	Unpack tar files		You should have six tar files	Run with UID=0

Step number and name	To-dos	Variables used	Validation	Comments
2.2 PTF2	Complete SMP/E steps for PTF2			Run with UID=0
2.3	Copy ejppiPTF.jcl to your install PDS lib <HLQ>.JOBS.CNTL.			
	Change all variables and submit ejppiPTF job	BBOLIB, #WPS# and #DRVNUM#	Check for rc=0	#WPS#: in JCL is the WPS HFS location (e.g. /usr/lpp/PortalServer); #DRVNUM# is 4.1.2
			See what directories have been created	
	Option: Use Oget to copy **all** the required JCLs from <WPS_PATH>/zosinst/jcl to <YourHLQ>.JOBS.CNTL now. Those JCLs will be needed for modification later.			EX: /usr/lpp/PortalServer/PortalServer/zosinst/jcl Run with UID=0
3. Save the log files	All jobs use and add to the same log file	<WPS_PATH>/Install.log		Make backups of log files for troubleshooting.
	This job use two log files WPSADMIN	<WPS_PATH>/zosinst/scripts/wps.admin.log <WPS_PATH>/Install.log		

Step number and name	To-dos	Variables used	Validation	Comments
	This job uses two log files WPACONF	<WPS_PATH> /temp/jobs/log/ WPConversation_date_time.log <WPS_PATH> /Install.log		
	This job use two log files WPSUNINS	<WPS_PATH> /WPConversation_date_time.log <WPS_PATH> /Install.log		
4. Edit Setup file	Oedit this setup file Refer to Appendix A.for environment variables	<WPS_PATH> zosinstall/scripts/wps.setup.in		Make backup of the original and save your changed setup to safe place UID=0 required
		<WPS_PATH> : Installation path of WebSphere Portal Server (WPS), not the HFS location which is referenced as <WPS> or #WPS#. Ex. WSPORT	Compare your variables with Appendix A	You should spend enough time to assure all the variables are entered correctly. Because it will be very time-consuming to troubleshoot those errors or make changes later. Some of the variables need to match with what your installation use. Ex: WebSphere, JDK, SysplexID and DB2 environment variables.
		<WPS_SERVERNAME>: WPS Server region name	The WPS Server region's STC name. Both step 5 (WLM) and step 6 (RACF) steps will use it	EX: <WPS_INSTALL_SYSTEMS>, <WPS_RESOURCE_LOCATION>, <WP_PATH>,<WS_CONFIG_PATH>, <DB2_PATH>, <DB2_PATH>,<JVM_LOG>

Step number and name	To-dos	Variables used	Validation	Comments
		<WPS_CTLREGIONID>: WPS control region Server userid		Step 6 uses it
		<WPS_PROCNAME>: WPS control region Procedure name. Ex: WSPORT	WPS Control region STC name.	WPS Server region Started Procedure name (STC) will have a suffix 'S' added Step (8.10) and step (11) use it.
		<WPS_GROUP>: Racf Group name for WPS . EX: WSPORTS		Need to match with Step (6) RACF definition.
		<WPS_LOGSTREAM>: EX: WAS.ERROR.LOG	Goto Syslog, issue /D LOGGER,L . The logstream name should exist already.	
		<WS_PATH>: Installation path for WebSphere AS		Need to match your WebSphere AS installation path.
		<JAVA_PATH>: It's should be /usr/lpp/java/IBM/J1.3	Validate the path	If does not exist, step(12) will have /usr/lpp/java/… not found error
5. Set up WLM		<WPS_SERVERNAME>, EX: WSPORT	Goto SDSF syslog and issue command: /D WLM,APPLENV=<Your_Applenv_name>	Note that the Server region Procedure name is the Control region name plus "S". Ex: WSPORTS

Step number and name	To-dos	Variables used	Validation	Comments
			WPS Server region procedure name must match with what WLM defines.	<Your_Applev_Name>: The WLM WPS Applenv name you selected.
6. Set up RACF	Verify all the related userids and groupids	<WPS_GROUP>, <WPS_SRVREGIONID>, WPS_CTLREGIONID>, <WPS_PROCNAME> and some your existing WebSphere RACF group names are used here too.	Server region procedure name should match with step(5) WLM setup.	Create a batch job RACF1 in <YourHLQ>.JOBS.CNTL and execute those RACF commands which are in /usr/lpp/PortalServer/PortalServer/zosinst/jcl/racf.jcl. However, make sure all the commands are executed successfully. The batch job always returns rc=0.
		The ADDUSER WSPORTSU line of the installation document has a typo error and will cause RACF syntax error on this line. Just delete the rest of the line starting from PW USER(WSPORTCU)… and replace it with NOPASSWORD.		No need to redo this step unless you have changed RACF variables in wps.setup.in file.

Step number and name	To-dos	Variables used	Validation	Comments
7. Configure LDAP Server	Start the WebSphere Customization Dialog	‘		This is a very critical step as it could overwrite your WebSphere Application Server customization datasets and saved CFG files.
	Issue the following command: exec '<BBO_Hlq>.SBBOCLIB(BBOWSTRT)'			
7.1	Choose option 1. New Customization.			
7.2	Load the saved customization variables for WebSphere Application Server.			The WPS LDAP customization need to be added to the existing WebSphere Application Server definitions.
7.3	Allocate a new set of target datasets. <YourHLQ>.CNTL and <YourHLQ>.DATA. Enter all the selected LDAP variables.	<WPS_LDAP_PORT>,EX: 2345, Enter the new LDAP Procedure name. EX: BBOLDAT		This will generate all the required JCLs and files in <YourHLQ>.CNTL and <YourHLQ>.DATA, which include those LDAP jobs BBOTDBMC, BBOLDGRT, and BBOLDAP.
7.5	Generate the customization jobs.		Will create two new <YourHLQ>.CNTL and <YourHLQ>.DATA datasets.	Make a backup of those three newly created datasets. This will help you troubleshooting your installation problem after it's modified later.

Step number and name	To-dos	Variables used	Validation	Comments
7.6	Save the customization variables to a new dataset. EX: TEST.PORTAL.WPS4.SAVECFG			Do not overwrite your existing WebSphere Application Server saved SAVECFG PDS. Document those three PDS datasets names.
7.8 (BBOLDAT)	Create the LDAP Procedure and copy them to SYS1.PROCLICLIB.			
7.9 (BBOPOLDF)	Create BBOPOLDF in <YourHLQ>.DATA Copy this file from <WPS_PATH>/zosinst/jcl and modify it. Change all dc=test,dc=com to your selected LDAP suffix. EX: "dc=ibm,dc=com". The <WPS_LDAP_ID> variable in the wps.setup.in file should have the same suffix.			This file will be copied to /tmp as /tmp/portal.ldif in step (7.11 and 7.15) by BBOWMCPT job. This is an ldif file format, be careful with the spaces. EX: one space is required after the ":" for each definition. One space is required for the next line in column 1 if the line continues. You should get familiar with some of the basics of LDAP such as ldif file format and how to prime the LDAP server, especially If you are running into problems at this step.

Step number and name	To-dos	Variables used	Validation	Comments
7.10 (BBOWTMCP)	Change <YourHLQ>.DATA(BBOWTMCP). In the first line, insert the "portal" after the bboslapd.config. Also insert the fifth line as shown in the Portal installation readme file.	<WS_CONFIG_PATH>: EX: /WebSphere390/CB390	Review your input typing and validate the <WS_CONFIG_PATH>/<YourSysplexID>/etc/ldap directory is exist.	This data file is set up for step (7.15) job BBOWMCPT to copy those five files listed on the left to the directory indicated on the right. You should make a backup of this directory (that is, <WS_CONFIG_PATH>/<YourSysplexID>/etc/ldap) and be very careful of not to caused it to overwrite the current bboslapd.conf files. They are used by your other LDAP servers.
7.11 (BBOLTDBM)	Change <YourHLQ>.DATA(BBOLTDBM). Note that there are five lines. Validate the " -f /tmp/portal.ldif" is part of the data required by the ldapmodify command. It's on the same line.			Member BBOLTDBM is used by step BBOMTDBM job to prime the LDAP TDBM database. The ldapmodify command is located in /usr/lpp/ldap/bin. It's used to prime those three listed ldif files to the new LDAP Server TDBM database. The fifth line has an "-a -h" instead of "-h" as the other lines. This means the ldapmodify command will add those entries in "/tmp/portal.ldif" to the LDAP TDBM.
7.12 (BBOWSLPT)	Change <YourHLQ>.DATA(BBOWSLPT). Make a backup of this file and change the securePort from 636 to a new one.			You can pick any number

Step number and name	To-dos	Variables used	Validation	Comments
	Remove the bottom sections of suffix and add "suffix "dc=<YOURDOMAIN>, dc=<YOURDOMAINSUFFIX>" to the bottom line.	Those suffix dc=<YOURDOMAIN> and dc=<YOURDOMAINSUFFIX> should match the suffix that you have selected in <WPS_LDAP_ID>. Also it is the same suffix as in step 7.19 and all the three ldif files listed in BBOWTMCP and later copied to /tmp as schema.user.ldif, schema.IBM.ldif and portal.ldif.		This suffix "dc=<YOURDOMAIN>, dc=<YOURDOMAINSUFFIX>" should match with the suffix specification in all the the three ldif files, /tmp/schema.user.ldif, /tmp/schema.IBM.ldif and portal.ldif used in step (7.15) BBOMTDBM job. This file will be copied to "<WPS_CONFIG_PATH>/YourSysplexID>/etc/ldap/<sysid>.bboslapd.conf.protal". It will be used as the //CONFIG DD reference file in LDAP Server STC (EX: BBOLDAT).

Step number and name	To-dos	Variables used	Validation	Comments
(BBOWULDF) (BBOWILDF)	Although it's not documented in the installation readme file, you should review and may need to modify those two members BBOWULDF and BBOWILDF its first line "dn: cn=schema, o=WASNaming, c=US" to match the suffix that you insert in step (7.12). Otherwise the schema.user.ldif and schema.IBM.ldif will not be primed. EX: "dc: cn=schema,dc=ibm,dc=com". No spaces between the ",".	Those suffix dc=<YOURDOMAIN> and dc=<YOURDOMAINSUFFIX> should match the suffix that you have selected in <WPS_LDAP_ID>.	Portal server currently only support two level index of the suffix. Errors could happen If you choose more than two level index.suffix.	If you have more than two levels of index in your Domain name, convert them to two level. EX: "uni.eng.man.edu", use "dc=uni.eng.man,dc=edu"
7.13	Run this second set of RACF commands. This RACF job sets up the necessary permissions for the new LDAP Server Procedure BBOLDAT.			Create this RACF2 job and save it in <YourHLQ>.JOBS.CNTL lib.
7.15	BBOLDTBD is only required if you want to delete the LDAP TDBM database due to problems.			Need DB2SYSADM Authority and WPS LDAP Server should be down.

Step number and name	To-dos	Variables used	Validation	Comments
(BBOTDBMC)	Create LDAP Server TDBM database			Need DB2SYSADM Authority and WPS LDAP Server should be down.
(BBOLDGRT)	Grant access to the LDAP database			Need DB2SYSADM Authority
(BBOWCPT)	Copy bboslapd.conf.portal to <WS_CONFIG_PATH>/SysplexID/etc/ldap. Copy all the three ldif files to /tmp directory.		Review the directory and its current LDAP conf files. Refer to step (7.10)	This is a very critical step as it could overwrite your exist LDAP Server configuration files. Make backup of this directory before running this job.
	Start WPS LDAP Server BBOLDAT.		Check joblog for message GLD0122I Slad is ready for requests.	
(BBOMTDBM)	This job primes the LDAP database with the ldapmodify commands. Refer to step (7.11)		Look for the sysout messages of "adding new entry dc=<YOURDOMAIN>, dc=<YOURDOMAINSUFFIX>. EX: adding new entry cn=users,dc=test,dc=com. All the new entries in portal.ldif should be added here and should be validated.	Use USER=CBADMIN All the new entries specified in portal.ldif should be added here. If not or job return code NE 0, then go back to step (7.9 to 7-12). You can also try to manually execute the ldapmodify commands under USS as listed in <YourHLQ>.DATA(BBOLTDBM) file. Make sure the LDAP Server is primed correctly before you continue.

Step number and name	To-dos	Variables used	Validation	Comments
	It executes the /tmp/schema.user.ldif first, then /tmp/schema.IBM.ldif next and /tmp/portal.ldif last. The schema.user.ldif need to be primed first.			You can use ldap browser (that is, DMT tool) to connect this LDAP Server and verify if all the portal.ldif entries are added.
8. Create the DB2 database for the Portal Server				This step starts the Portal Server configuration.
(CREATEDB)	Create database for Portal Server. For DB2 V7.1, the plan name is "DSNTIA71". Add LIB(<DB2HLQ>.RUNLIB.LOAD) for locating this plan.	Replace #DB2PRFX#, the DB2 HLQ, #DB2SUB#, the DB2 Subsystem name, #PLAN# the DB2 plan name, #WPS# the WPS HFS location, #YOURHLQ# the <YourHLQ>.JOBS.CNTL. Do this for the four jobs.		You may need to add the UNIT=VIO to your sysprint DD for the first job step of the job.
(INITTABL)	Initialize DB2 fields	See above		
(INITDB)	Create fields and perform DB2 grant.	See above		

Step number and name	To-dos	Variables used	Validation	Comments
(DROPDB)	Drop database. Only needed if this step need to be rerun.	See above		Resolve all the problems before continue. #WPS_SRVREGIONID# should be the same as specified in wps.setup.in (step 4).
Create WSPORT and WSPORTS Procedures	Create both procedures and copy them to SYS1.PROCLIB	Control Region Server name <WPS_PROC NAME> and Server region name in WLM.		Review all the Portal Server related RACF definitions.
9. Configure WPS on WebSphere Application Server	Oget all the 5 jcl's from <WPS_PATH>/z osinst/jcl to <YourHLQ>.JO BS.CNTL			Portal Server WSPORT should be down.
	Replace the following variables for all five files: <BBOLIB>, #WPS#, HFS location of the WPS and #INPUTFILE#.		If you have moved the wps.setup.in to a new location, then #INPUTFILE# is the new location of the wps.setup.in file. Delete it if you choose the default location	
(WPSPRE)	Replace variables and submit the job		Review joblog	Uid=0 required. If job fails, fix the problem and rerun the job. All the shell scripts are located in <WPS_PATH>/zosinst/script s directory. Review those scripts and compare it with the error messages of the job output.

Step number and name	To-dos	Variables used	Validation	Comments
(WPSADMIN)	WebSphere Application Server need to be up. Replace variables and submit the job.		Review Joblog	Use USER=CBADMIN Use TIME=1440 This job also need more time to run.
(WPSPOST)	Replace variables and submit the job.		Review joblog	Use UID=0 The shell script wpspost.root.in uses grep cmd to locate both 'wpsPre.root.sh finished successfully' and wpsAdmin.root.sh finished successfully' messages from the <WPS_PATH>/Install.log before continue.
11. Start WebSphere Portal				
11.2	Issues the command S <WPS_PROCNAME>.<WPS_SERVERNAMEA>. EX: /S WSPORT.WSPORTA Point your browser to: http//<WPS_HOST>:<WPS_PORT>/wps/<WPS_PUBLIC_HOME>. EX: http://here.there.everywhere:8082/wps/portal		You will see a page with a message stating that your portal contains no page groups yet. Logon to WebSphere Portal as the user wpsadmin and verify that the login was successful.	This complete the Portal Server Configuration.
12. Deploy the base portlets				WPS should be running.

Step number and name	To-dos	Variables used	Validation	Comments
(WPAINS)	Replace variables and #JOBCOUNT#. Submit the job.		WebSphere Portal Server should be up.	Use UID=0 Resolve all problems before continue.
(WPACONF)	Replace variables and #JOBCOUNT#. submit the job.			Use UID=0 Resolve all problems before continue. WebSphere Portal Server should be down.
12.8	Restart the Portal Server			
	Point your browser to: http//<WPS_HOST>:<WPS_PORT>/wps/<WPS_PUBLIC_HOME>. EX: http://here.there.everywhere:8082/wps/Portal		You should see the complete Portal Welcome page.	If fails, determine which steps may have had problem and restart from that step. Restart from step (4), (7), (8) or (9) accordingly.

Note: After any changes, should you need to delete a conversation you can use the `<yourHLQ>.jobs.cntl(wpsunions)`. This job runs the shell script `WPUninstall` to create a new conversation under SMEUI, deletes Portal and its datasource, and then commits the conversation. See step two of Uninstalling PTF2 in the README file for further information.

3

Development environment installation

In this chapter we present these topics:

- The software provided on CD-ROMs for development of portlets and content authoring.
- Installation of WebSphere Studio Application Developer and the Portlet Toolkit.
- Installation of WebSphere Portal Content Publishing, its authoring and development environment.

3.1 Software provided on CD-ROM

The WebSphere Portal development environment is provided on a set of CD-ROMs for installation on a distributed platform based on the availability of software for that platform, as follows:

1. Portlet development and debugging and WebSphere Portal Content Publishing wizards
   - WebSphere Studio Application Developer V4.0.3, for Microsoft Windows
   - WebSphere Portal Toolkit, for Microsoft Windows
   - WebSphere Application Server Advanced Edition Single Server V4.0.2, for Microsoft Windows
   - IBM DB2 Universal Edition V7.2 with Fix Pack 7, for Multiplatforms
   - WebSphere Portal Enable V4.1.3, for Microsoft Windows
2. WebSphere Portal Content Publishing authoring
   - WebSphere Portal Content Publishing V4.2 (author-time), for Multiplatforms
   - WebSphere Application Server Advanced Edition V4.0.4, for Multiplatforms
   - IBM DB2 Universal Edition V7.2 with Fix Pack 7, for Multiplatforms
   - IBM Lotus Domino Application Server V5.0.10, for Multiplatforms
   - IBM Lotus Workflow and Lotus Workflow Architect 3.0a, for Multiplatforms

Table 3-1 shows a complete list of the CD-ROMs provided and their content.

Table 3-1 CD-ROM contents

CD-ROM	Software
1	DB2 UDB Enterprise Edition for Windows (SBCS)
2	DB2 UDB Enterprise Edition for Windows (DBCS)
3	DB2 UDB Enterprise Edition for AIX (SBCS)
4	DB2 UDB Enterprise Edition for AIX (DBCS) Japan/Korea
5	DB2 UDB Enterprise Edition for AIX (DBCS) Chinese
6	DB2 UDB Enterprise Edition for Solaris (SBCS)
7	DB2 UDB Enterprise Edition for Solaris (DBCS) Japan/Korea
8	DB2 UDB Enterprise Edition for Solaris (DBCS) Chinese

CD-ROM	Software
9	DB2 UDB Enterprise Edition for Linux (SBCS/DBCS)
10	DB2 UDB Fixpack 7 for Windows
11	DB2 UDB Fixpack 7 for AIX
12	DB2 UDB Fixpack 7 for Solaris
13	DB2 UDB Fixpack 7 for Linux
14	DB2 UDB Enterprise Edition for Linux on zSeries (SBCS/DBCS)
15	DB2 UDB Fixpack for Linux on zSeries (SBCS/DBCS)
16	WebSphere Application Server AE for AIX and Solaris
17	WebSphere Application Server AE for Windows and Linux
18	WebSphere Application Server AEs V402 for Windows and Portal Toolkit for Windows
19	WebSphere Application Server AE for Linux for zSeries
20	Secureway Directory for AIX, Solaris, Windows and Linux
21	Secureway Directory for Linux for zSeries
22	WebSphere Studio Application Developer for Windows
23	WebSphere Portal Server Enable
24	Lotus Domino Application Server for AIX and Solaris
25	Lotus Domino Application Server for Windows and Linux
26	Lotus Domino Application Server languages
27	Lotus Domino Application Server languages
28	Lotus Domino Application Server languages
29	Lotus Domino Application Server for Windows
30	Lotus Domino Application Server for Windows
31	Lotus Domino Application Server for Windows
32	Lotus Notes Client Domino Designer and Administrator for Windows (SBCS)
33	Lotus Notes Client Domino Designer and Administrator for Windows (DBCS)

CD-ROM	Software
34	Infocenter documentation for WebSphere Portal on Multiplatforms
35	Lotus Workflow
36	WebSphere Portal Content Publishing

Figure 3-1 on page 97 shows a simple example of a development and authoring environment for WebSphere Portal on z/OS that could be used in a Proof of Concept. This is example is taken from the WebSphere Portal Content Publishing (WPCP) Help System available at:

http://publib.boulder.ibm.com/wcmid/mp/v42/helpsystem/en/index.html

In the example, WebSphere Studio Application Developer is installed on a Microsoft Windows 2000 platform with the Portlet Toolkit plug-in for portlet development, and also the WPCP authoring wizards plug-in. The server in the Development environment generates and delivers content, Web pages (HTML/JSPs), servlets, portlets, application objects, resource classes, and content spots to servers in both the Authoring and Production environments.

Additionally the WPCP Authoring Server is installed on another Microsoft Windows 2000 platform for business users and other content contributors. They can create and publish content to the Web site on the Production Server via the WebSphere Portal content publishing workspace.

Finally in this example, WebSphere Portal and WPCP runtime are installed and running on a z/OS server in the Runtime environment as the Production server.

Further examples can be found in the WPCP Help System that make use of Lotus Workflow Architect on the Development Server, Lotus Domino Workflow on the Authoring Server, IBM Content Manager and Likeminds Recommendation Engine on the z/OS Production Server.

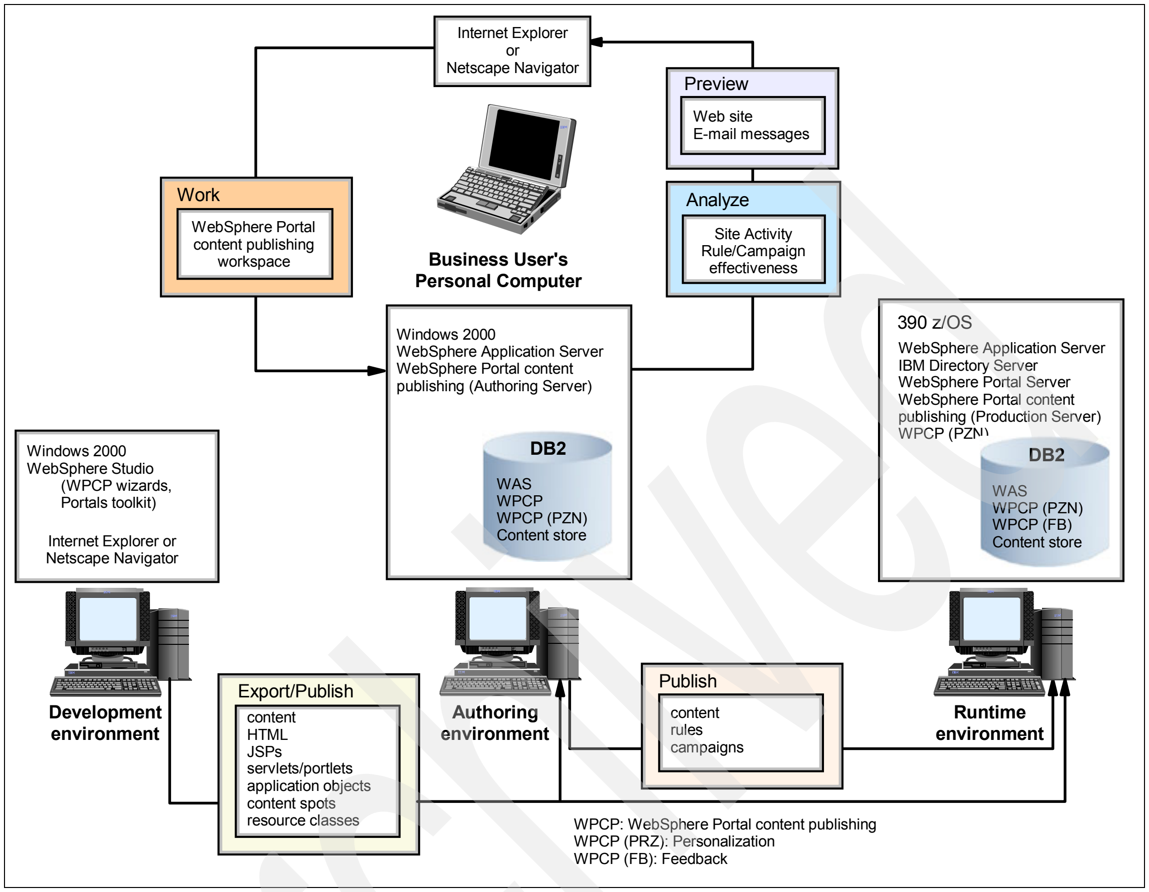


Figure 3-1 Simple development environment for WebSphere Portal on z/OS

3.2 Portlet development and debugging

WebSphere Portal Server is a J2EE Web application running on WebSphere Application Server, and provides a Portlet API for development of Java Web applications, called portlets, that have access to the services provided by the J2EE container, classes and tag libraries provided with WebSphere Application Server. We recommend users new to developing portlets refer to the following redbook publications to get started with the design and development of portal applications.

A Portal Composite Pattern Using WebSphere V4.1, SG24-6869, provides design patterns, information and guidelines for the development of portlet applications, including examples for WebSphere Portal.

WebSphere Studio Application Developer Programing Guide, SG24-6585, provides detailed information on developing Web applications using WebSphere Studio Application Developer.

3.2.1 Installing the Portal Toolkit with Application Developer

The IBM WebSphere Developer Domain library contains many useful articles and papers on developing portlets using the Portal Toolkit with WebSphere Studio Application Developer (Application Developer). We used two papers to guide us with the installation of the Portal Toolkit and WebSphere Studio Application Developer, including installation of the debugging environment using WebSphere Application Server Advanced Edition Single Server and WebSphere Portal Enable 4.1. In the following sections we describe the installation we performed.

For useful information, see the IBM WebSphere Developer Domain library at:

http://www7b.software.ibm.com/wsdd/

Installing prerequisite software

We recommend that you refer to the paper by Doug Phillips entitled *Installing Prerequisite Software for the IBM Portal Toolkit Plug-in for WebSphere Studio Application Developer,* which describes the installation of the following components on a Microsoft Windows 2000 platform. You can find this paper at:

http://www7b.software.ibm.com/wsdd/library/techarticles/0210_phillips/phillips2.html

- WebSphere Application Server Advanced Edition Single Server
- IBM DB2
- WebSphere Portal Enable 4.1
- WebSphere Studio Application Developer

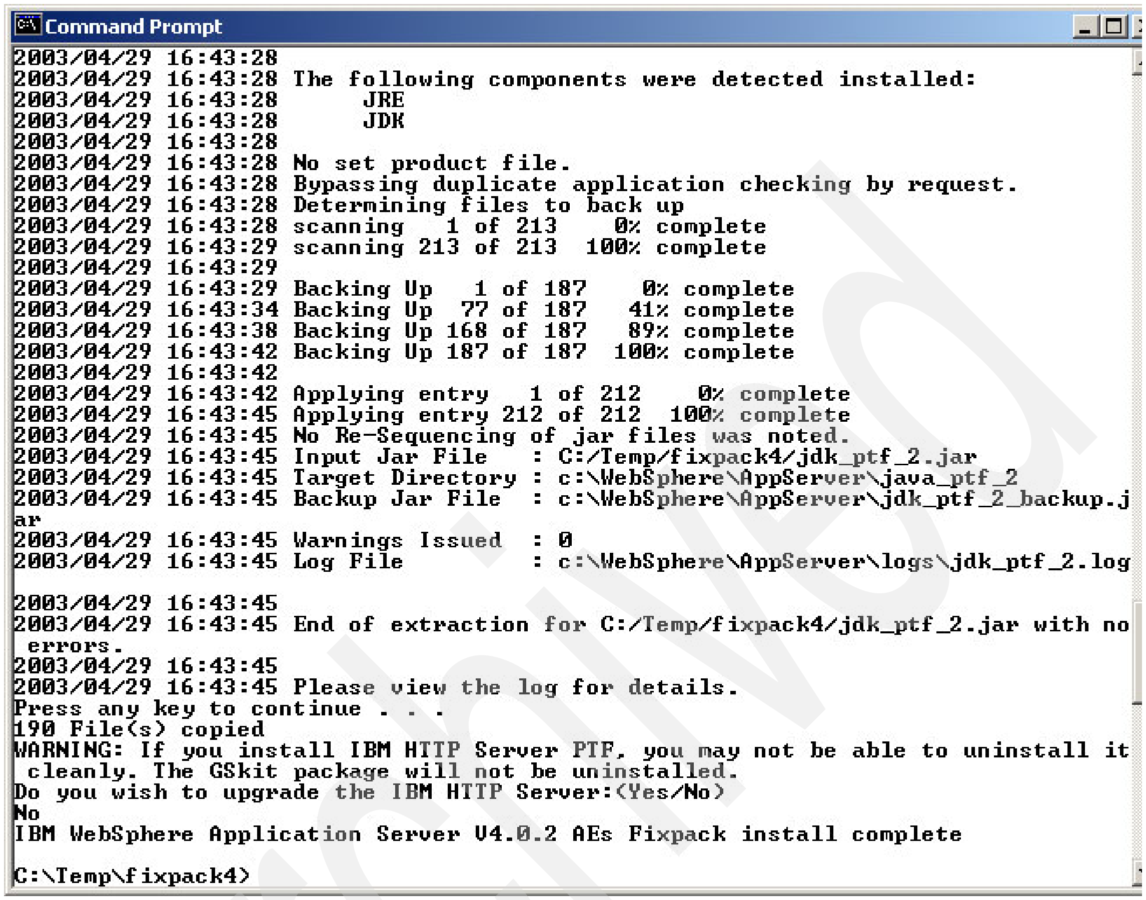
Using this paper as our guide we performed the following steps making note here of the differences we encountered by using the software supplied on the CD-ROMs for WebSphere Portal for z/OS:

Install WebSphere Application Server

We used CD-ROM number 18 and started the installation by running `setup.exe` from directory \was\win. (Refer to Table 3-1 on page 94 for CD-ROM contents and their corresponding numbers.)

Install WebSphere Application Server Fixpack 2

Fixpack 2 is contained on CD-ROM 18, but rather confusingly in directory \was\win\fixpack4. We followed the rest of the instructions in the paper for applying the fixpack. Figure 3-2 shows the successful installation.



```
2003/04/29 16:43:28
2003/04/29 16:43:28 The following components were detected installed:
2003/04/29 16:43:28      JRE
2003/04/29 16:43:28      JDK
2003/04/29 16:43:28
2003/04/29 16:43:28 No set product file.
2003/04/29 16:43:28 Bypassing duplicate application checking by request.
2003/04/29 16:43:28 Determining files to back up
2003/04/29 16:43:28 scanning   1 of 213    0% complete
2003/04/29 16:43:29 scanning 213 of 213  100% complete
2003/04/29 16:43:29
2003/04/29 16:43:29 Backing Up   1 of 187    0% complete
2003/04/29 16:43:34 Backing Up  77 of 187   41% complete
2003/04/29 16:43:38 Backing Up 168 of 187   89% complete
2003/04/29 16:43:42 Backing Up 187 of 187  100% complete
2003/04/29 16:43:42
2003/04/29 16:43:42 Applying entry   1 of 212    0% complete
2003/04/29 16:43:45 Applying entry 212 of 212  100% complete
2003/04/29 16:43:45 No Re-Sequencing of jar files was noted.
2003/04/29 16:43:45 Input Jar File    : C:/Temp/fixpack4/jdk_ptf_2.jar
2003/04/29 16:43:45 Target Directory : c:\WebSphere\AppServer\java_ptf_2
2003/04/29 16:43:45 Backup Jar File   : c:\WebSphere\AppServer\jdk_ptf_2_backup.j
ar
2003/04/29 16:43:45 Warnings Issued   : 0
2003/04/29 16:43:45 Log File          : c:\WebSphere\AppServer\logs\jdk_ptf_2.log

2003/04/29 16:43:45
2003/04/29 16:43:45 End of extraction for C:/Temp/fixpack4/jdk_ptf_2.jar with no
 errors.
2003/04/29 16:43:45
2003/04/29 16:43:45 Please view the log for details.
Press any key to continue . . .
190 File(s) copied
WARNING: If you install IBM HTTP Server PTF, you may not be able to uninstall it
 cleanly. The GSkit package will not be uninstalled.
Do you wish to upgrade the IBM HTTP Server:(Yes/No)
No
IBM WebSphere Application Server V4.0.2 AEs Fixpack install complete

C:\Temp\fixpack4>
```

Figure 3-2 WebSphere Application Server Fixpack2 installation

Install e-fix PQ56615

To run WebSphere Portal on WebSphere Application Server V402, e-fix PQ56615 needs to be installed and this e-fix was also available on CD-ROM number 18. Figure 3-3 on page 100 shows the successful installation of PQ56615.

Copy Personalization files needed by WebSphere Portal

WebSphere Portal needs personalization jar files:

- personalisation.jar
- prCommon.jar

We copied these from CD-ROM number 36, directory \wpcp\wcp\runtime\lib.

Install DB2 version 7.2

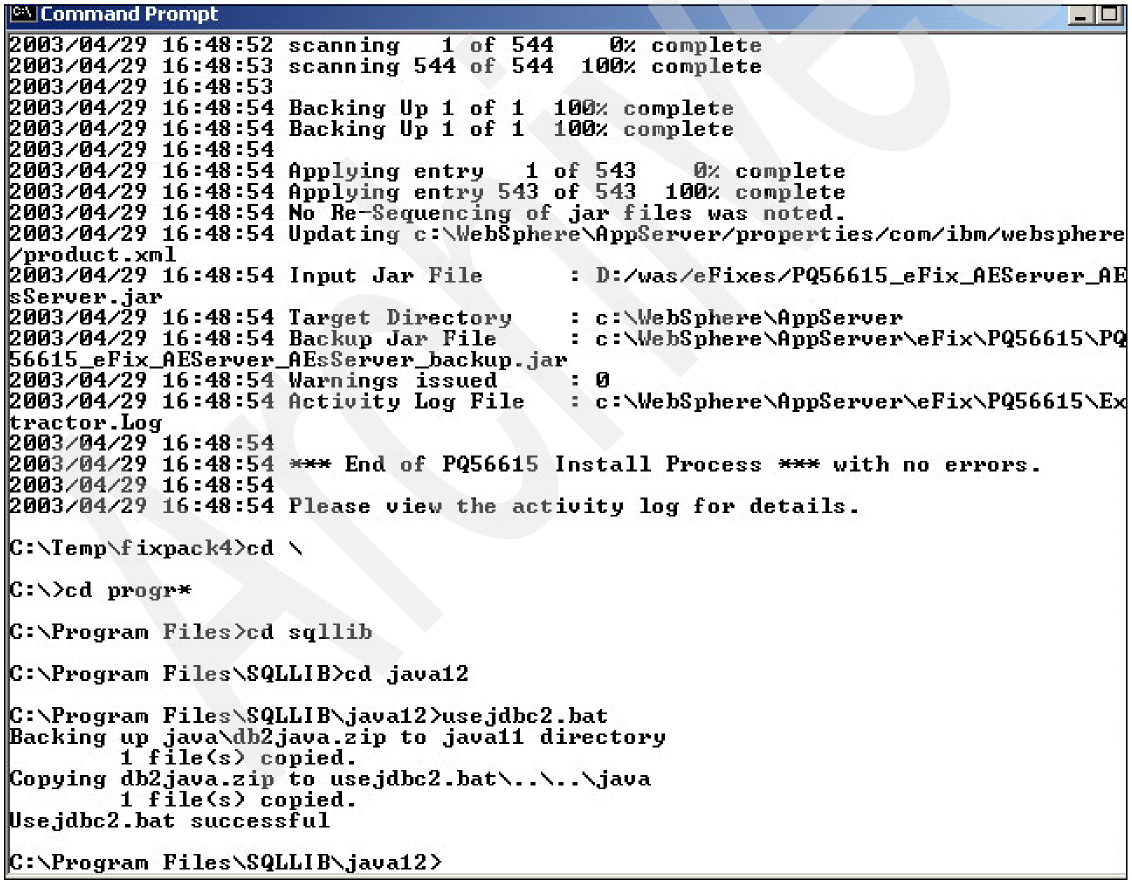
We installed DB2 Enterprise Edition V7.2 using CD-ROM number 1 by running `setup.exe` from directory \db2\win and following the instructions in the paper (the paper we referred to by Doug Phillips; see “Installing prerequisite software” on page 98 for information).

Install DB2 fixpack 7

CD-ROM number 10 contains DB2 fixpack 7 which we installed instead of fixpack 5 (referred to in the paper by Doug Phillips). We installed fixpack 7 by running `setup.exe` from directory \db2fp\win.

Configure the JDBC driver to use Java 1.2

Figure 3-3 shows the successful completion of configuring the JDBC driver in accordance with the instructions in the paper.

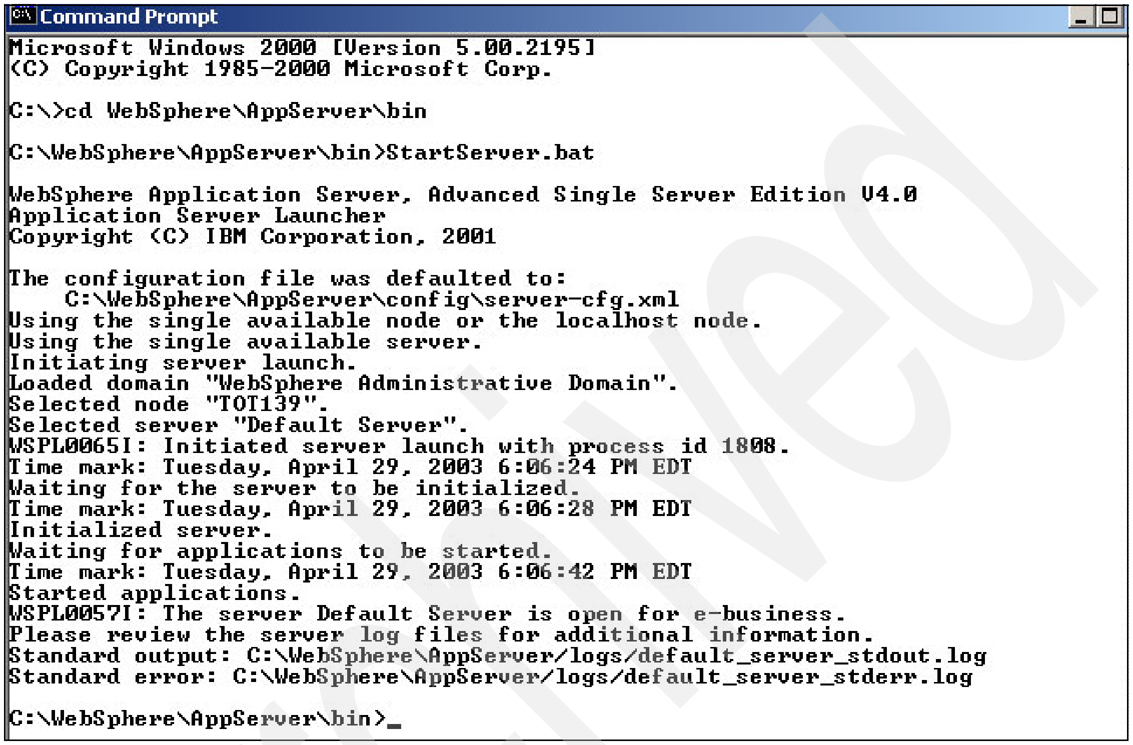


```
Command Prompt
2003/04/29 16:48:52 scanning   1 of 544    0% complete
2003/04/29 16:48:53 scanning 544 of 544  100% complete
2003/04/29 16:48:53
2003/04/29 16:48:54 Backing Up 1 of 1  100% complete
2003/04/29 16:48:54 Backing Up 1 of 1  100% complete
2003/04/29 16:48:54
2003/04/29 16:48:54 Applying entry   1 of 543    0% complete
2003/04/29 16:48:54 Applying entry 543 of 543  100% complete
2003/04/29 16:48:54 No Re-Sequencing of jar files was noted.
2003/04/29 16:48:54 Updating c:\WebSphere\AppServer/properties/com/ibm/websphere
/product.xml
2003/04/29 16:48:54 Input Jar File      : D:/was/eFixes/PQ56615_eFix_AEServer_AE
sServer.jar
2003/04/29 16:48:54 Target Directory    : c:\WebSphere\AppServer
2003/04/29 16:48:54 Backup Jar File     : c:\WebSphere\AppServer\eFix\PQ56615\PQ
56615_eFix_AEServer_AEsServer_backup.jar
2003/04/29 16:48:54 Warnings issued     : 0
2003/04/29 16:48:54 Activity Log File   : c:\WebSphere\AppServer\eFix\PQ56615\Ex
tractor.Log
2003/04/29 16:48:54
2003/04/29 16:48:54 *** End of PQ56615 Install Process *** with no errors.
2003/04/29 16:48:54
2003/04/29 16:48:54 Please view the activity log for details.

C:\Temp\fixpack4>cd \

C:\>cd progr*

C:\Program Files>cd sqllib

C:\Program Files\SQLLIB>cd java12

C:\Program Files\SQLLIB\java12>usejdbc2.bat
Backing up java\db2java.zip to java11 directory
        1 file(s) copied.
Copying db2java.zip to usejdbc2.bat\..\..\java
        1 file(s) copied.
Usejdbc2.bat successful

C:\Program Files\SQLLIB\java12>
```

Figure 3-3 Successful installation of PQ56615 and configuring JDBC driver

Install WebSphere Portal Enable V4.1

After starting WebSphere Application Server as shown in Figure 3-4 we used CD-ROM number 25 to run `install.bat` from directory \wps, in order to install WebSphere Portal Server.

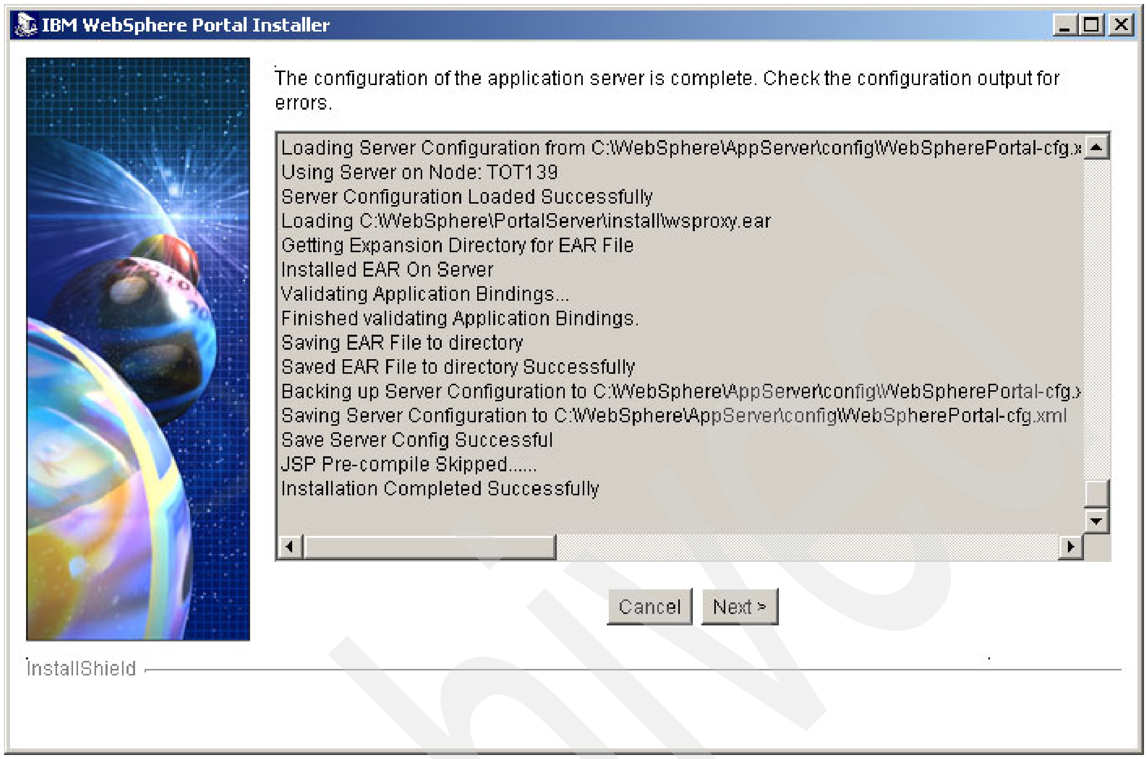


```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>cd WebSphere\AppServer\bin

C:\WebSphere\AppServer\bin>StartServer.bat

WebSphere Application Server, Advanced Single Server Edition V4.0
Application Server Launcher
Copyright (C) IBM Corporation, 2001

The configuration file was defaulted to:
    C:\WebSphere\AppServer\config\server-cfg.xml
Using the single available node or the localhost node.
Using the single available server.
Initiating server launch.
Loaded domain "WebSphere Administrative Domain".
Selected node "TOT139".
Selected server "Default Server".
WSPL0065I: Initiated server launch with process id 1808.
Time mark: Tuesday, April 29, 2003 6:06:24 PM EDT
Waiting for the server to be initialized.
Time mark: Tuesday, April 29, 2003 6:06:28 PM EDT
Initialized server.
Waiting for applications to be started.
Time mark: Tuesday, April 29, 2003 6:06:42 PM EDT
Started applications.
WSPL0057I: The server Default Server is open for e-business.
Please review the server log files for additional information.
Standard output: C:\WebSphere\AppServer/logs/default_server_stdout.log
Standard error: C:\WebSphere\AppServer/logs/default_server_stderr.log

C:\WebSphere\AppServer\bin>_
```

Figure 3-4 Starting WebSphere Application Server Advanced Edition Single Server

We followed all the steps, 1 to 28, in the paper noting the following details as we proceeded:

- Step 16, select “Create and initialize a **new** database”.
- Step 25, we noticed and ignored some error messages “Failed reflecting values” during the deployment of the portal enterprise applications. The successful completion of this step is shown in Figure 3-5 on page 102.



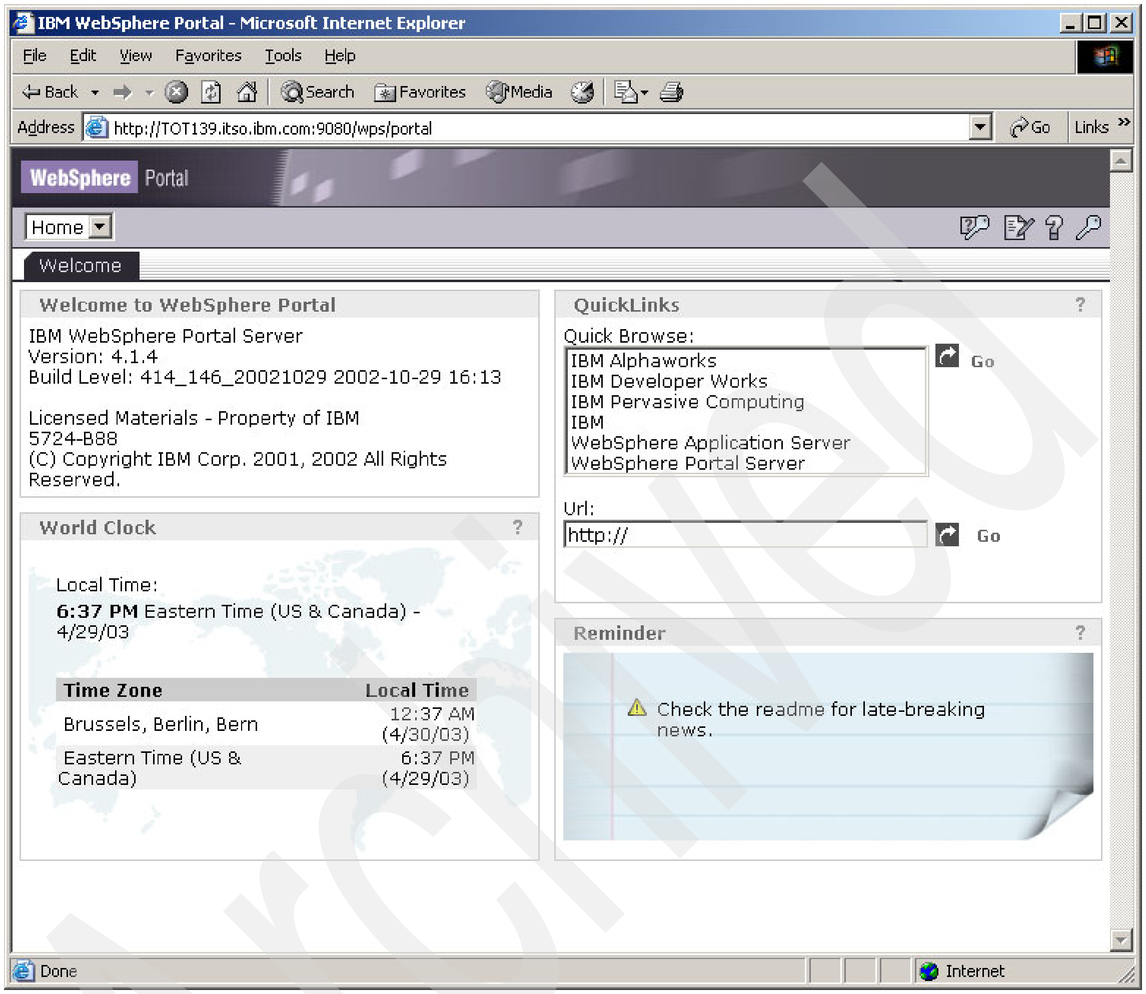
The configuration of the application server is complete. Check the configuration output for errors.

```
Loading Server Configuration from C:\WebSphere\AppServer\config\WebSpherePortal-cfg.x
Using Server on Node: TOT139
Server Configuration Loaded Successfully
Loading C:\WebSphere\PortalServer\install\wsproxy.ear
Getting Expansion Directory for EAR File
Installed EAR On Server
Validating Application Bindings...
Finished validating Application Bindings.
Saving EAR File to directory
Saved EAR File to directory Successfully
Backing up Server Configuration to C:\WebSphere\AppServer\config\WebSpherePortal-cfg.)
Saving Server Configuration to C:\WebSphere\AppServer\config\WebSpherePortal-cfg.xml
Save Server Config Successful
JSP Pre-compile Skipped......
Installation Completed Successfully
```

Figure 3-5 Installation of WebSphere Portal

- On completion of Step 28 we tested that the portal was installed successfully and operational (Figure 3-6 on page 103) by browsing to:

  `http://ourhost:9080/wps/portal`


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://TOT139.itso.ibm.com:9080/wps/portal
Go Links
WebSphere Portal
Home
Welcome

Welcome to WebSphere Portal

IBM WebSphere Portal Server
Version: 4.1.4
Build Level: 414_146_20021029 2002-10-29 16:13

Licensed Materials - Property of IBM
5724-B88
(C) Copyright IBM Corp. 2001, 2002 All Rights Reserved.

World Clock

Local Time:
6:37 PM Eastern Time (US & Canada) - 4/29/03

Time Zone	Local Time
Brussels, Berlin, Bern	12:37 AM (4/30/03)
Eastern Time (US & Canada)	6:37 PM (4/29/03)

QuickLinks

Quick Browse:
IBM Alphaworks
IBM Developer Works
IBM Pervasive Computing
IBM
WebSphere Application Server
WebSphere Portal Server

Url:
http://

Reminder

Check the readme for late-breaking news.

Figure 3-6 Accessing the portal from a browser

(Optional) Installing the Agent Controller

We did not install the Agent Controller, because in the next steps we will install WebSphere Studio Application Developer on the same system as WebSphere Application Server and WebSphere Portal.

Install WebSphere Studio Application Developer

We used CD-ROM number 22 and ran `setup.exe` from directory \wsad\win to install WebSphere Studio Application Developer.

Installing the Portal Toolkit

Doug Phillips wrote an additional paper on installing the Portal Toolkit, configuring it to use the installed Portal Server for debugging, and developing a first portlet to test the set-up. This paper, *Developing and Debugging Portlets Using the IBM Portal Toolkit Plug-in for WebSphere Studio Application Developer,* is available at:

```
http://www7b.software.ibm.com/wsdd/library/techarticles/0210_phillips/phillips.
html
```

Using this paper as our guide we performed the following steps:

- Registered userid `wpsdebug` with the Portal Server.
- Stopped WebSphere Application Server.
- Used CD-ROM number 18 to run `install.bat` from directory \PortalToolkit to install the Portal Toolkit.
- Configured the portal server instance in WebSphere Studio, taking note that on Step 6 of the paper you must un-check the box labelled "Use default WebSphere deployment directory", and on Step 13 change to port 9080.
- Created MyFirstPorlet, published it to the portal Debug Server, and tested it by following Steps 1 through 6 in the paper.

Conclusion

Upon completion of these steps the portlet development environment was installed and configured to debug and test portlets using the WebSphere Portal Server installed on the Windows 2000 platform. Further information about developing portlets using WebSphere Studio Application Developer can be found in the *IBM WebSphere Portal Developers Handbook,* SG24-6897.

Portlets that have been developed using the portlet development environment can be exported as Web Application Archive (WAR) files from WebSphere Studio Application Developer as shown in Figure 3-7 on page 105 — by right-clicking on the project name, selecting **Export WAR**, and then transferring the WAR file to WebSphere Portal running on z/OS for subsequent deployment.

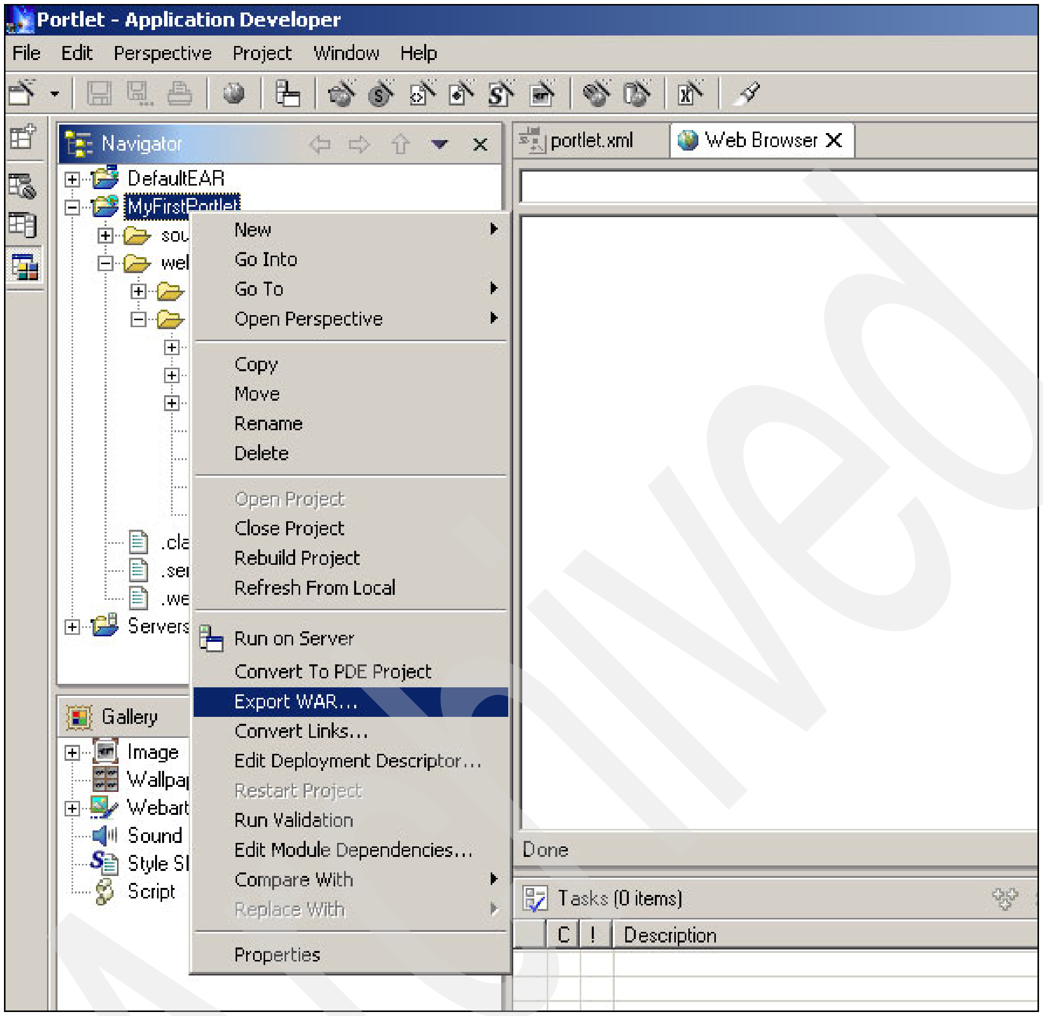


Figure 3-7 Exporting a WAR file

3.3 WebSphere Portal Content Publishing

The WebSphere Portal Content Publishing (WPCP) system for z/OS and its prerequisites can be divided into three functional environments:

1. WPCP runtime environment:
   - Installed on WebSphere Portal for z/OS as described in Chapter 2, “Portal installation” on page 19.

2. WPCP authoring environment consisting of:
   - WPCP authoring server installed on a supported distributed platform.
3. WPCP development consisting of:
   - Wizards provided for WebSphere Studio Application Developer installed on a Microsoft platform.

The remainder of this chapter deals with the installation of the WPCP authoring and development environments on a distributed platform.

The README and installation guide for WPCP is contained in directory \wpcp\doc on CD-ROM number 36. Further information is available as follows:

- The Getting Started guide at:

  `http://publib.boulder.ibm.com/wcmid/mp/v42/helpsystem/en/gspdf.pdf`
- The help system at:

  `http://publib.boulder.ibm.com/wcmid/mp/v42/helpsystem/en/index.html`

3.3.1 Installing the WPCP authoring environment

The WPCP authoring environment is installed on WebSphere Application Server, Advanced Edition 4.0.3 or later, and uses either a DB2 or Oracle database, or the IBM Content Manager, as a data repository. User authentication is done via LDAP or the operating system registry.

Workflow for the authoring environment can be done with either Lotus Workflow installed on Domino Application Server and managed using Lotus Workflow Architect, or Lite Workflow which is provided as part of WPCP.

To install the WPCP authoring environment using the software provided on CD-ROMs with WebSphere Portal on z/OS we installed the following components on a Windows 2000 system:

- DB2 Enterprise Edition V7.2 with Fixpack 7
- WebSphere Application Server Advanced Edition V4.0.4
- WPCP authoring server using Lite Workflow

We performed the following steps to install the authoring environment on a Windows 2000 system.

Install DB2 version 7.2

We installed DB2 Enterprise Edition using CD-ROM number 1 by running `setup.exe` from directory \db2\win.

Install DB2 fixpack 7

CD-ROM number 10 contains DB2 fixpack 7 which we installed by running `setup.exe` from directory \db2fp\win.

Configure the JDBC driver to use Java 1.2

Figure 3-3 on page 100 shows the successful completion of configuring the JDBC by running `usejdbc2.bat`.

WebSphere Application Server Advanced Edition

We used CD-ROM number 17 to install WebSphere Application Server Advanced Edition and ran `setup.exe` from directory \was\win.

We chose a typical install, which installs IBM HTTP Server, WebSphere Application Server, the Administrative Server, the Administration Console, and JDK 1.3.0.

WebSphere PTF4

CD-ROM number 17 contains WebSphere PTF4 in directory \fixpack4 and we followed the instructions in the directory to install the PTF by running `install.bat`.

Additional WebSphere fixes

CD-ROM number 17 contains four additional fixes in directory \was\efixes, that we installed by following the instructions in the README supplied for each fix.

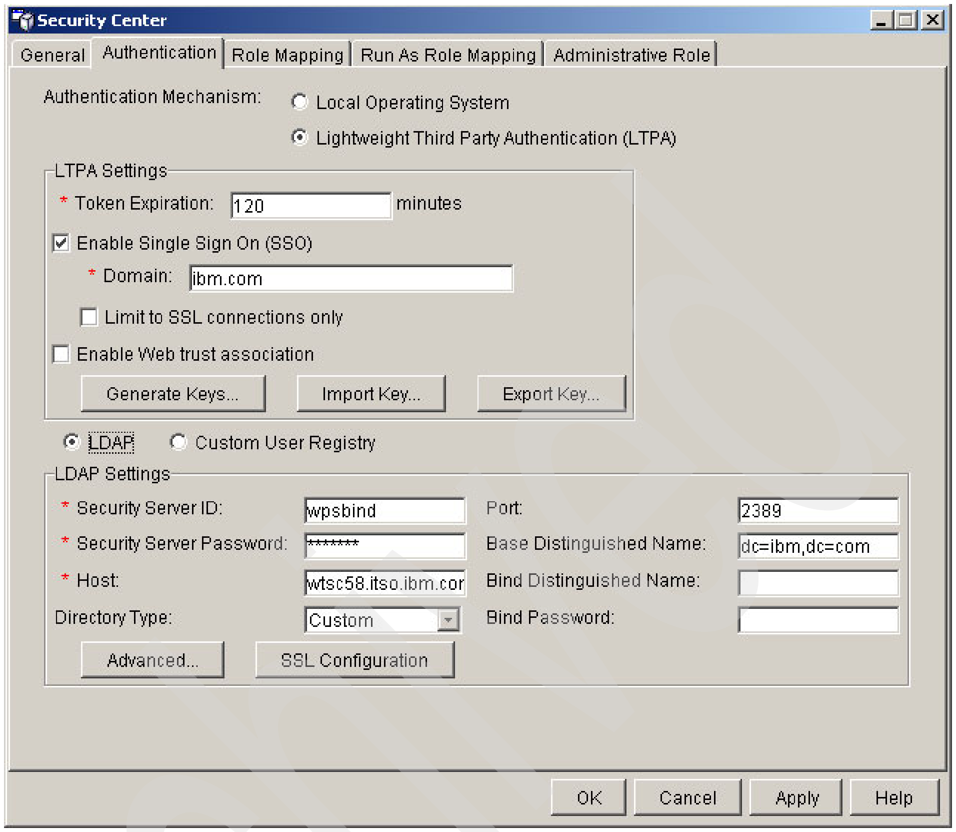
Enabling WebSphere security

On completion of installing WebSphere Application Server, PTF4, and the additional fixes, we re-booted the system, started the Administrative Server and Default Server, and then tested whether it was working by browsing to:

```
http://localhost/servlet/snoop
```

We decided to enable security from the WebSphere Security Console at this stage (because WPCP will need it, once it is installed) and configure the WebSphere Application Server to use the LDAP running on our z/OS system for authentication.

Figure 3-8 on page 108 shows the settings for Authentication in the Security Console specifying our LDAP server on z/OS. Note that we specified a `Custom Directory Type`, port `2389`, suffix `dc=ibm,dc=com` and host `wtsc58.itso.ibm.com`® which was our z/OS system for WebSphere Portal.



Authentication Mechanism: ( ) Local Operating System [✓ Lightweight Third Party Authentication (LTPA)]

LTPA Settings

* Token Expiration: 120 minutes

[✓] Enable Single Sign On (SSO)

* Domain: ibm.com

[ ] Limit to SSL connections only

[ ] Enable Web trust association

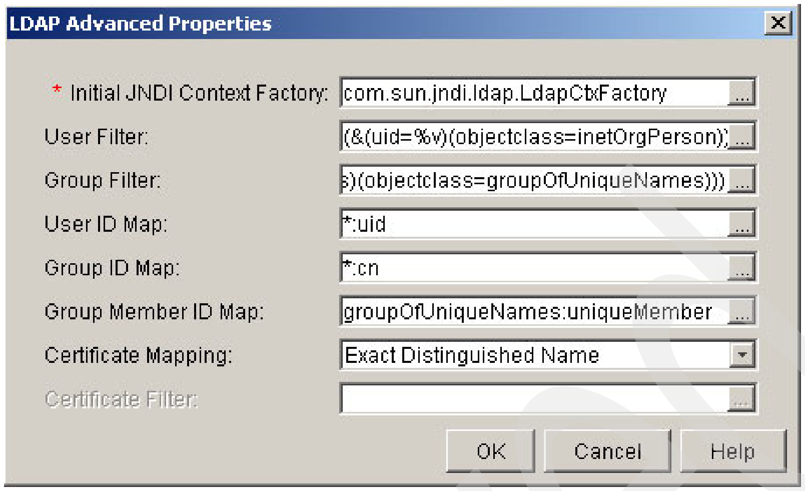
[✓ LDAP] ( ) Custom User Registry

LDAP Settings

* Security Server ID:	wpsbind	Port:	2389
* Security Server Password:	*******	Base Distinguished Name:	dc=ibm,dc=com
* Host:	wtsc58.itso.ibm.cor	Bind Distinguished Name:	
Directory Type:	Custom	Bind Password:	

Figure 3-8 Security console settings for Authentication

Finally, because we have specified a custom directory, we clicked on the **Advanced** button and specified `inetOrgPerson` in the User Filter as shown in Figure 3-9 on page 109.



* Initial JNDI Context Factory:	com.sun.jndi.ldap.LdapCtxFactory
User Filter:	(&(uid=%v)(objectclass=inetOrgPerson))
Group Filter:	s)(objectclass=groupOfUniqueNames)))
User ID Map:	*:uid
Group ID Map:	*:cn
Group Member ID Map:	groupOfUniqueNames:uniqueMember
Certificate Mapping:	Exact Distinguished Name
Certificate Filter:	

Figure 3-9 LDAP settings WPCP Authoring Server

Installing the WPCP Authoring Server

We started the installation of the WPCP Authoring Server by running `wcpinstallwin.exe` from directory \wpcp\wcpinstallwin.exe on CD-ROM number 36.

Because we had enabled security in the WebSphere Application Server, we were prompted for the security userid and password and we logged in.

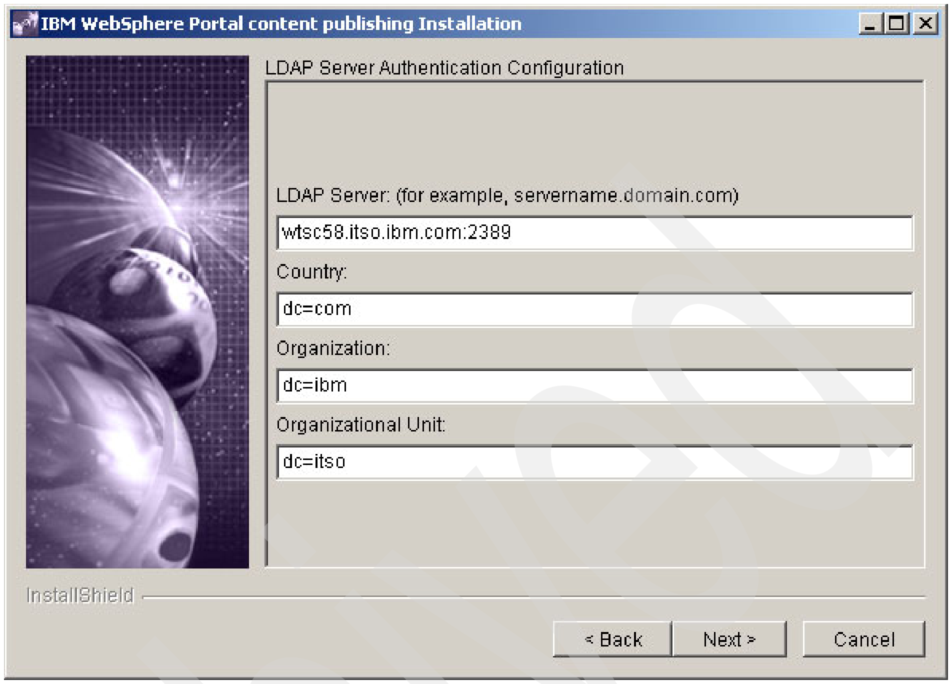
From the installation options screen we selected to install the Authoring Server and Sample.

We selected to install the Authoring Server on the WebSphere Application Server `Default Server` and `default_host`.

We selected a Database as the repository and not IBM Content Manager.

For workflow, we selected Lite Workflow, because this is the simplest configuration that does not require installation of Domino Application Server, Lotus Workflow, and Lotus Workflow Architect.

For the authentication mechanism we chose LDAP with settings as shown in Figure 3-10 on page 110.



LDAP Server Authentication Configuration

LDAP Server: (for example, servername.domain.com)
wtsc58.itso.ibm.com:2389

Country:
dc=com

Organization:
dc=ibm

Organizational Unit:
dc=itso

Figure 3-10 WPCP settings for LDAP

Tip 1: The suffix settings do not get recorded correctly in the configuration file used to store them and will need to be corrected in the next step after installation.

Tip 2: The release of WebSphere Portal for z/OS, that we tested with, did not support three fields in the LDAP suffix, so although we have entered dc=itso,dc=ibm,dc=com we need to change this after the installation as well.

We chose the option to pre-compile JSPs and then completed the installation.

Configuring and testing the WPCP Authoring Server

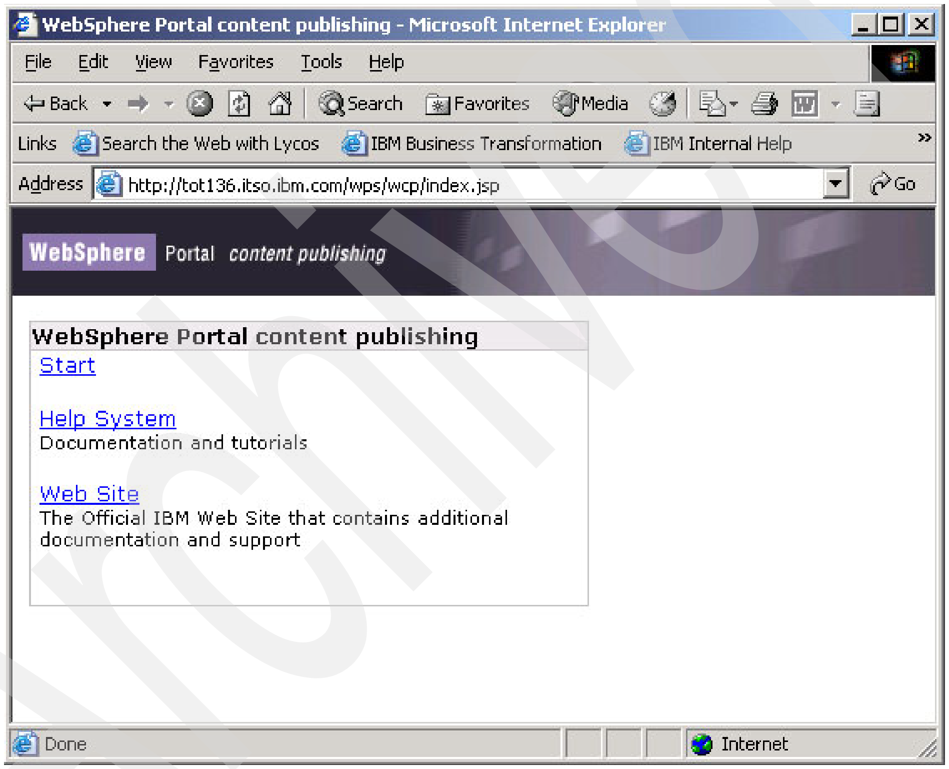
We performed the following post-installation configuration steps:

- Edited <WAS_ROOT>\wcp\author\WCM.properties to correct the LDAP suffix as follows: `wcm.workflow.ldapBase=dc=ibm,dc=com`. Note that we have changed this to use only two fields in the suffix.
- Added users `rob`, `tara`, `greg`, `dave` and `WCPAdmin` used by the WPCP sample to the LDAP server running on z/OS.

Tip: We added users by using the Directory Management Tool that comes with IBM Directory Server and which can be installed from CD-ROM number 20. Alternatively, if WebSphere Portal on z/OS is up and running, then an easy way to do this is by adding users from a browser through the portal interface.

After stopping and restarting the `default_server` application server, where we had installed WPCP Authoring Server, we used a browser to access the WPCP system (Figure 3-11) by going to:

```
http://<ourhost>/wps/wcp/index.jsp
```



WebSphere Portal content publishing - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://tot136.itso.ibm.com/wps/wcp/index.jsp
Go
WebSphere Portal content publishing
Done
Internet

WebSphere Portal content publishing

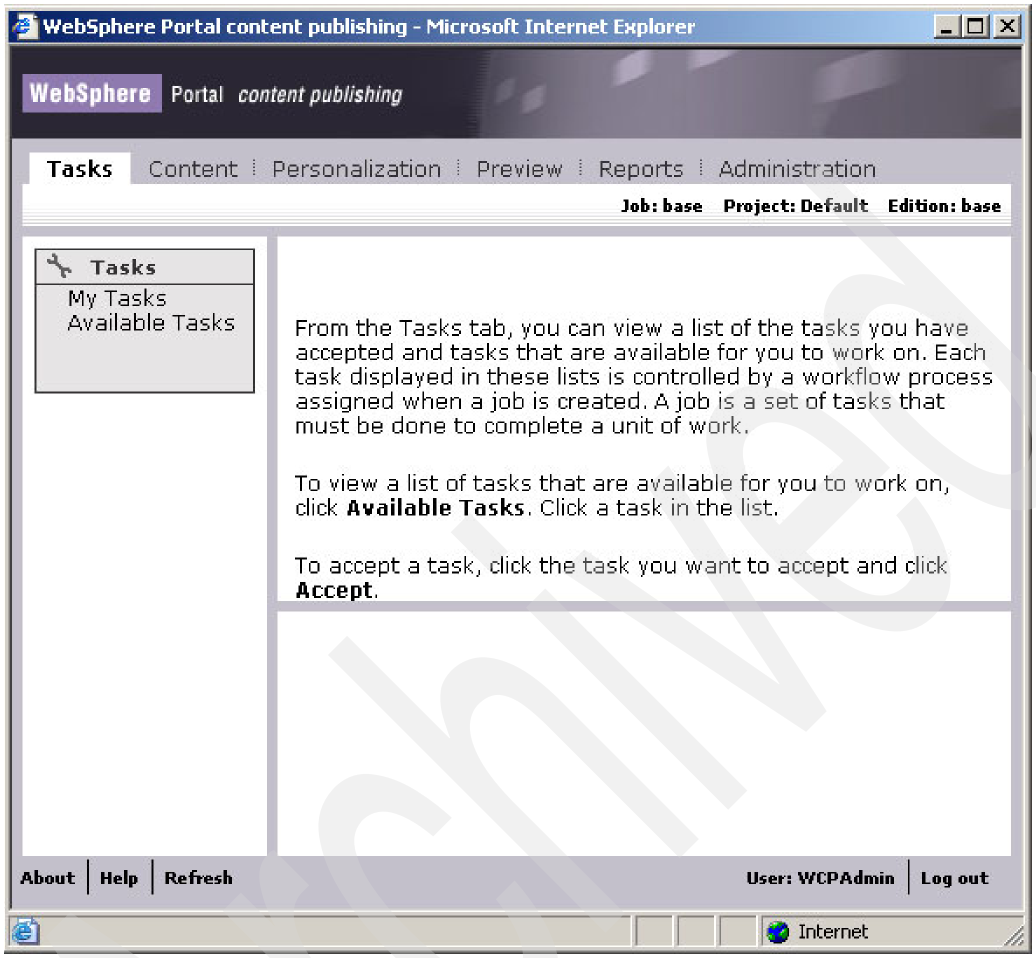
Start

Help System
Documentation and tutorials

Web Site
The Official IBM Web Site that contains additional documentation and support

Figure 3-11 WPCP index

By clicking **Start** and logging in with userid `WCPAdmin` we accessed the WPCP administrators panel as shown inFigure 3-12 on page 112.



From the Tasks tab, you can view a list of the tasks you have accepted and tasks that are available for you to work on. Each task displayed in these lists is controlled by a workflow process assigned when a job is created. A job is a set of tasks that must be done to complete a unit of work.

To view a list of tasks that are available for you to work on, click **Available Tasks**. Click a task in the list.

To accept a task, click the task you want to accept and click **Accept**.

Figure 3-12 WPCP Administration page

Installation of the WPCP Authoring environment is now complete and we recommend that users new to WPCP refer to the *Getting Started Tutorial* available from the Help System.

Testing the runtime publishing server

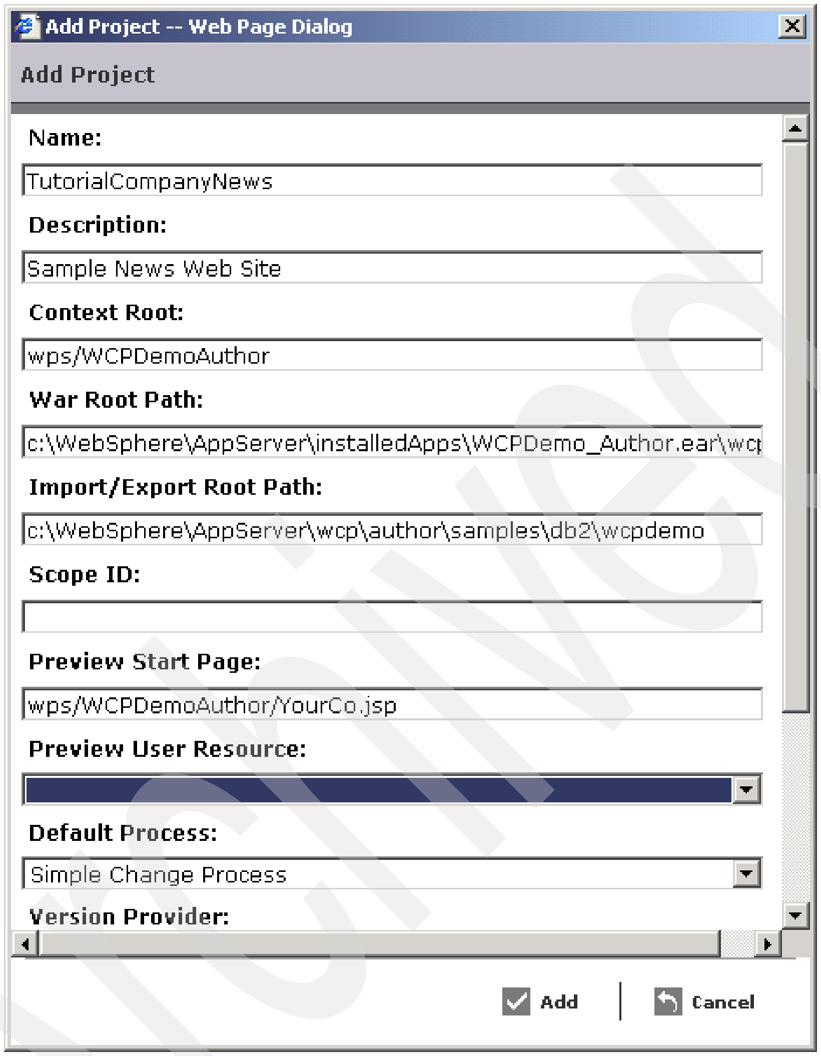
Having installed WPCP authoring server on our development system, we wanted to test whether we could publish to the WPCP publisher runtime server running on z/OS. The WPCP publisher runtime installation is covered in 2.7, “WebSphere Portal Content Publishing PTF2” on page 64.

The *Getting Started Tutorial* available from the WPCP Help System describes a tutorial that can be used with a sample demo that is installed with WPCP authoring. We decided to use this demo, create a project, and publish it to our runtime server on z/OS.

We do not repeat all of the steps from the tutorial here — creating and adding new content, managing workflow between different users — but performed the basic steps needed to create and publish a project. These are the key steps we performed, taken from the tutorial, as follows:

Tutorial step: Setting up the project

We added the WCPDemo example to a new project in our Authoring Server by following the help instructions (Figure 3-13 on page 114).


Add Project -- Web Page Dialog

Add Project

Name:
TutorialCompanyNews

Description:
Sample News Web Site

Context Root:
wps/WCPDemoAuthor

War Root Path:
c:\WebSphere\AppServer\installedApps\WCPDemo_Author.ear\wc

Import/Export Root Path:
c:\WebSphere\AppServer\wcp\author\samples\db2\wcpdemo

Scope ID:

Preview Start Page:
wps/WCPDemoAuthor/YourCo.jsp

Preview User Resource:

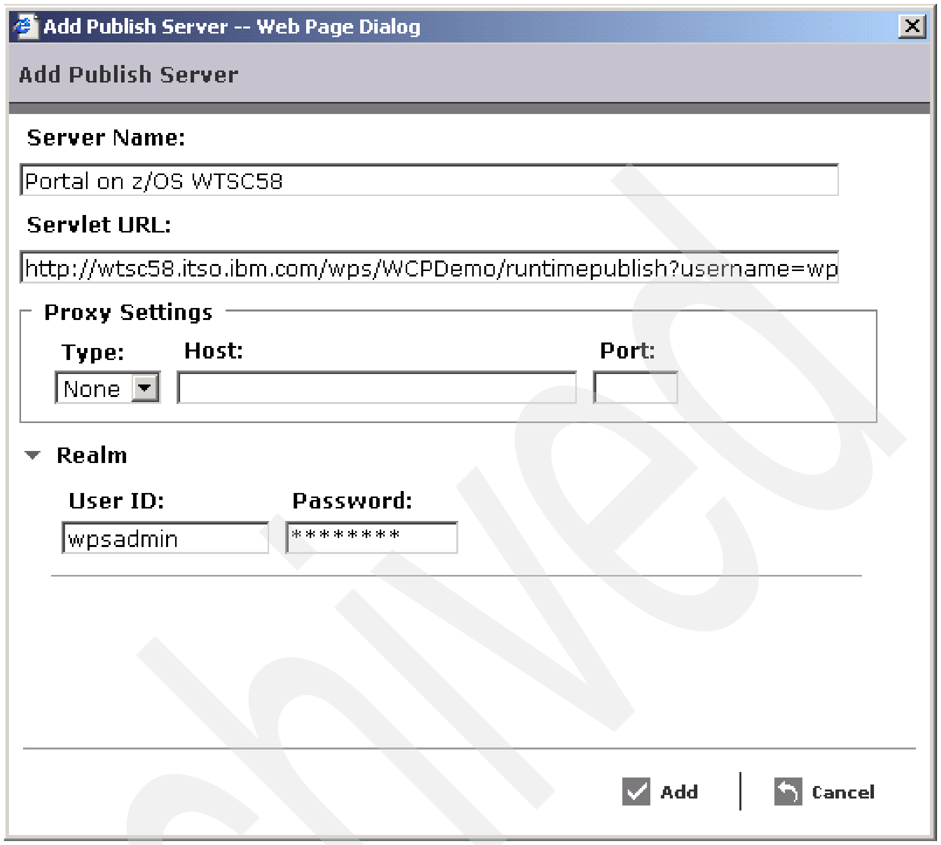
Default Process:
Simple Change Process

Version Provider:

Figure 3-13 Add the sample demo project to WPCP authoring server

Tutorial step: Setting up a publish server

In this step we defined our z/OS WPCP publisher runtime server, that was installed in 2.7, “WebSphere Portal Content Publishing PTF2” on page 64, to the project as shown in Figure 3-14 on page 115.


Add Publish Server -- Web Page Dialog

Add Publish Server

Server Name: Portal on z/OS WTSC58

Servlet URL: http://wtsc58.itso.ibm.com/wps/WCPDemo/runtimepublish?username=wp

Proxy Settings

Type: None

Host:

Port:

Realm

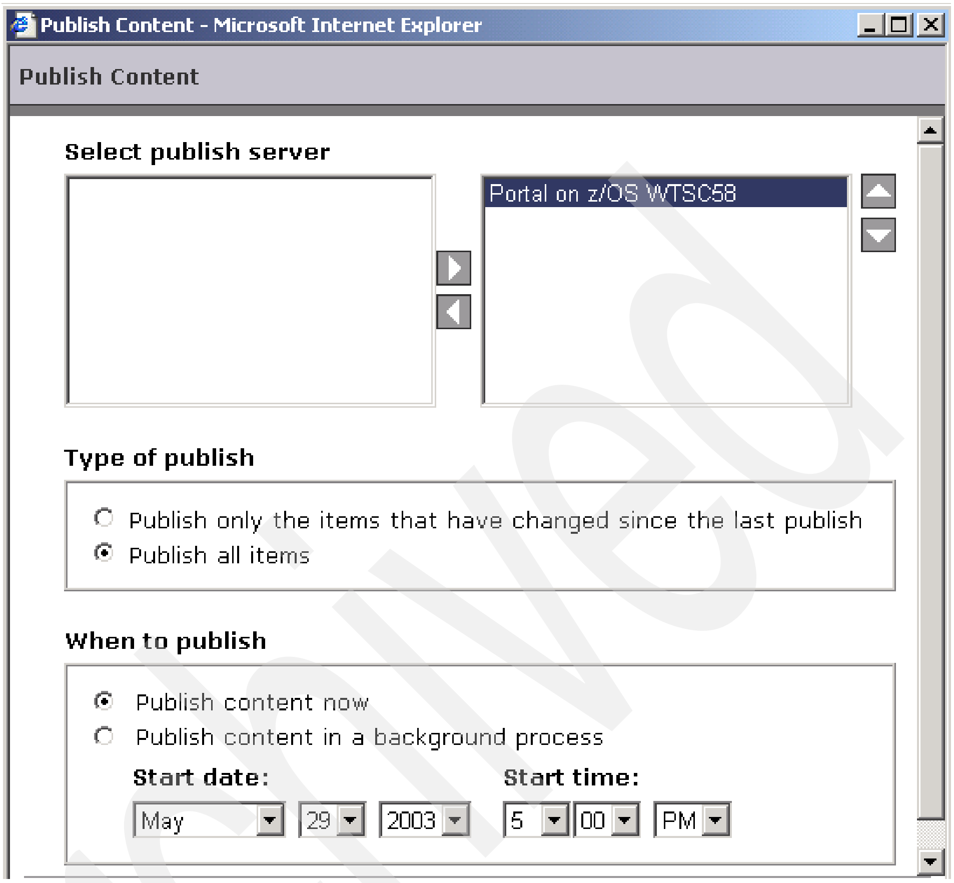
User ID: wpsadmin

Password: ********

Figure 3-14 Defining the z/OS WPCP publishing server to the authoring server project

Tutorial step: Publishing the project

The next to last step is to actually publish the project from the Authoring server to the Publisher runtime server on z/OS as shown in Figure 3-15 on page 116.



Publish Content

Select publish server

Portal on z/OS WTSC58

Type of publish

( ) Publish only the items that have changed since the last publish
(•) Publish all items

When to publish

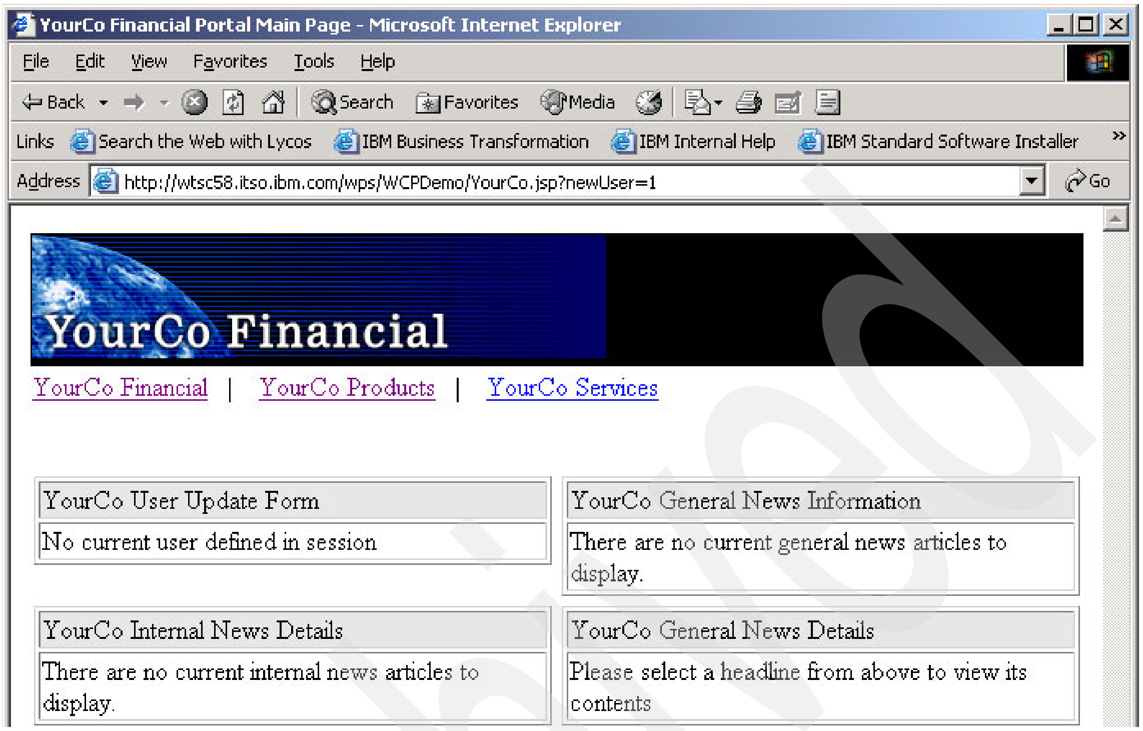
(•) Publish content now
( ) Publish content in a background process

Start date: May 29 2003

Start time: 5 00 PM

Figure 3-15 Publishing the project to z/OS Portal system

After publishing the project to the WPCP runtime server our final check was to verify that the server was operational and users were able to access the content published from the project. The tutorial describes that this is done by accessing the URL `http://<yourruntimeserver>/wps/WCPDemo/YourCo.jsp`, which we did as shown in Figure 3-16 on page 117.


YourCo Financial Portal Main Page - Microsoft Internet Explorer

Address: http://wtsc58.itso.ibm.com/wps/WCPDemo/YourCo.jsp?newUser=1

YourCo Financial

YourCo Financial | YourCo Products | YourCo Services

YourCo User Update Form
No current user defined in session

YourCo General News Information
There are no current general news articles to display.

YourCo Internal News Details
There are no current internal news articles to display.

YourCo General News Details
Please select a headline from above to view its contents

File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Links
Search the Web with Lycos
IBM Business Transformation
IBM Internal Help
IBM Standard Software Installer
Go

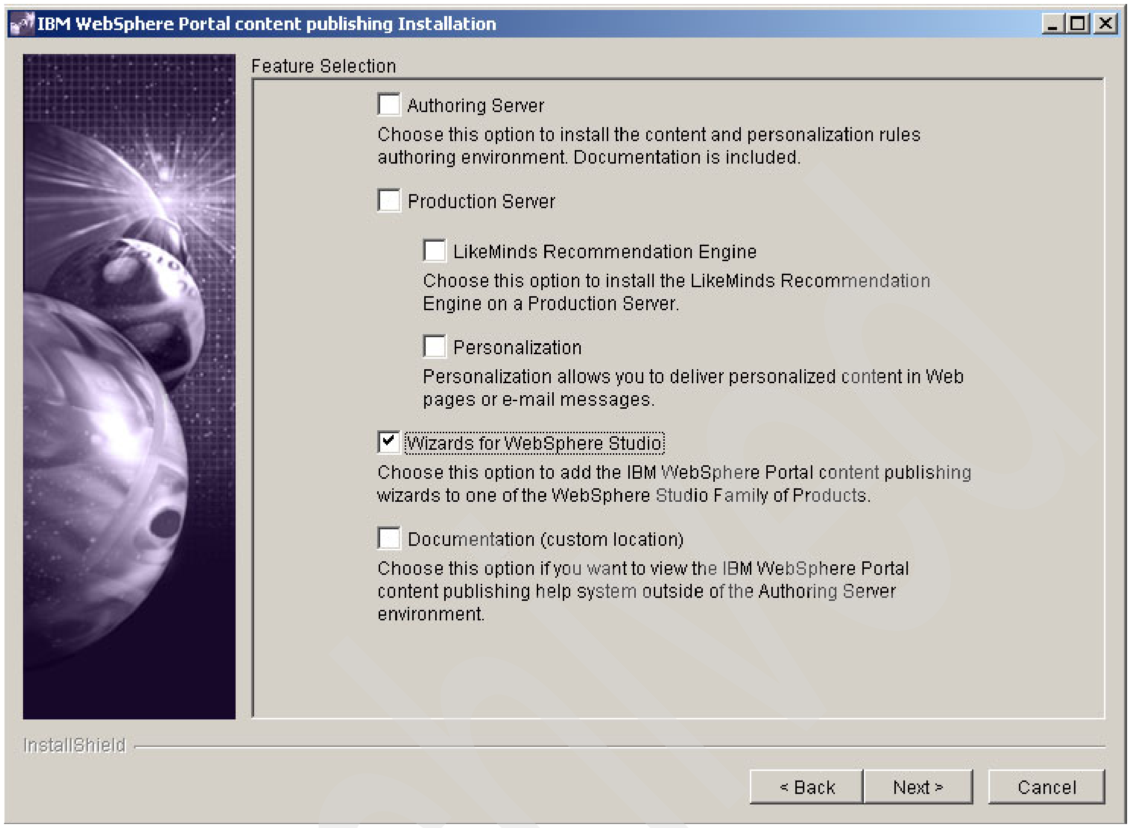
Figure 3-16 WPCP runtime publisher for sample project

Although the displayed published project content is devoid of any information, that was merely because we had not followed all the tutorial steps for adding content to the project from the demo. However, we have proved that we can author content in our development environment and publish it to the z/OS WPCP runtime environment.

3.3.2 Installing the WPCP development environment

To install the WPCP development environment, consisting of plug-ins for WebSphere Studio Application Developer we performed the following steps using the same Windows 2000 system used in 3.2, “Portlet development and debugging” on page 97.

1. Using CD-ROM number 36 we ran `wcpinstallwin.exe` from directory \wpcp.
2. Selected the WebSphere Studio wizards to be installed as shown in Figure 3-17 on page 118 and completed the installation using the install wizard.


IBM WebSphere Portal content publishing Installation

Feature Selection

[ ] Authoring Server
Choose this option to install the content and personalization rules authoring environment. Documentation is included.

[ ] Production Server

[ ] LikeMinds Recommendation Engine
Choose this option to install the LikeMinds Recommendation Engine on a Production Server.

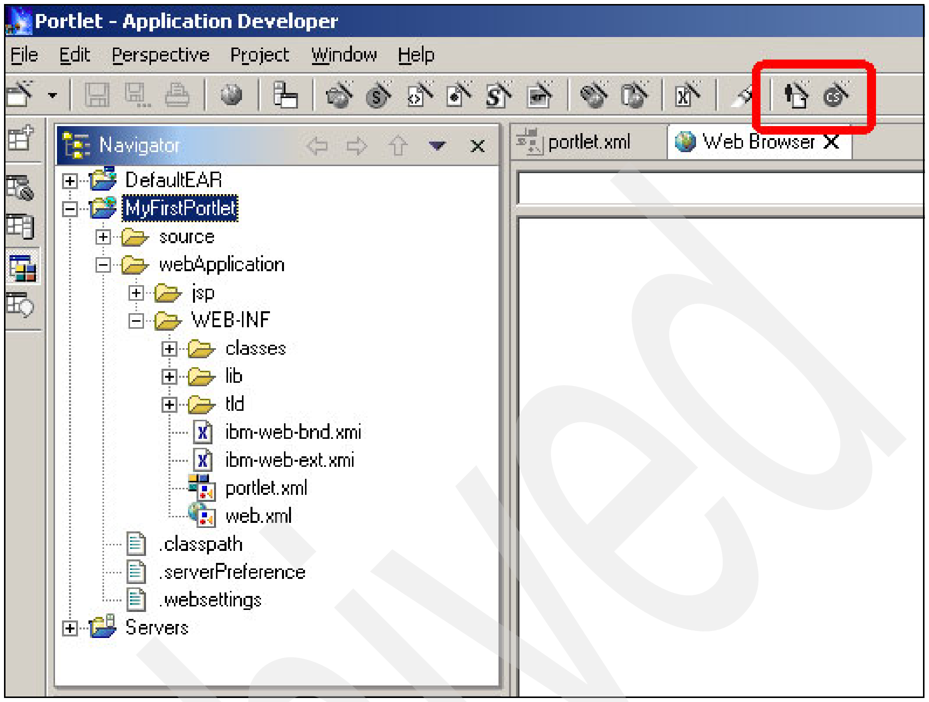
[ ] Personalization
Personalization allows you to deliver personalized content in Web pages or e-mail messages.

[✓] Wizards for WebSphere Studio
Choose this option to add the IBM WebSphere Portal content publishing wizards to one of the WebSphere Studio Family of Products.

[ ] Documentation (custom location)
Choose this option if you want to view the IBM WebSphere Portal content publishing help system outside of the Authoring Server environment.

Figure 3-17 Wizard selection for WebSphere Studio

3. Verified the installation by checking that the following folders had been created:
   – C:\Program Files\IBM\WebSphereStudio\Application Developer\plugins\com.ibm.wcm.resource.wizards
   – C:\Program Files\IBM\WebSphereStudio\Application Developer\plugins\com.ibm.wcm.resource.wizards.doc
4. Opened WebSphere Studio Application Developer, clicked on **Perspective**-> **Customize**, then expanded **Other** and selected **WebSphere Portal content publishing toolbar**, and then clicked **OK**.
5. This results in two WPCP wizards shown in the toolbar of the active Perspective, as shown in Figure 3-18 on page 119.


Portlet - Application Developer
File Edit Perspective Project Window Help
Navigator
portlet.xml
Web Browser

```
DefaultEAR
MyFirstPortlet
    source
    webApplication
        jsp
        WEB-INF
            classes
            lib
            tld
            ibm-web-bnd.xmi
            ibm-web-ext.xmi
            portlet.xml
            web.xml
    .classpath
    .serverPreference
    .websettings
Servers
```

Figure 3-18 WPCP wizards installed for WebSphere Studio

The WPCP development environment for WebSphere Studio was now installed and we recommend that users new to WPCP refer to the section on Wizards in the *Developers Guide* in the WPCP Help System.

4

Portal administration

In this chapter we introduce some of the WebSphere Portal administration functions. We also cover functions like creating places and pages, that are typically available not only to the portal administrator, but also to other portal users. Setting Access Control Lists (ACLs) is touched upon, but a deeper discussion about security and the architecture surrounding portal security on z/OS takes place in Chapter7, “Portal security” on page 283 of this book.

4.1 Administrative privileges

When WebSphere Portal is installed, the default portal administrator with userid and password of `wpsadmin` is created. The other default user that gets created is `wpsbind`. The `wpsbind` user is used by WebSphere to bind to the LDAP server. Remember both these users are part of the `wpsadmins` group. If you want any other user to have portal administrative authority, that user should be a member of the `wpsadmins` group. Of course it is possible to change these userids, but we suggest that you wait to do this until after the installation is complete.

How can you tell the difference between an ordinary user and one who has administrative authority? Look at the Portal place menu options. In a Web browser bring up the default base WebSphere Portal page by entering the following URL:

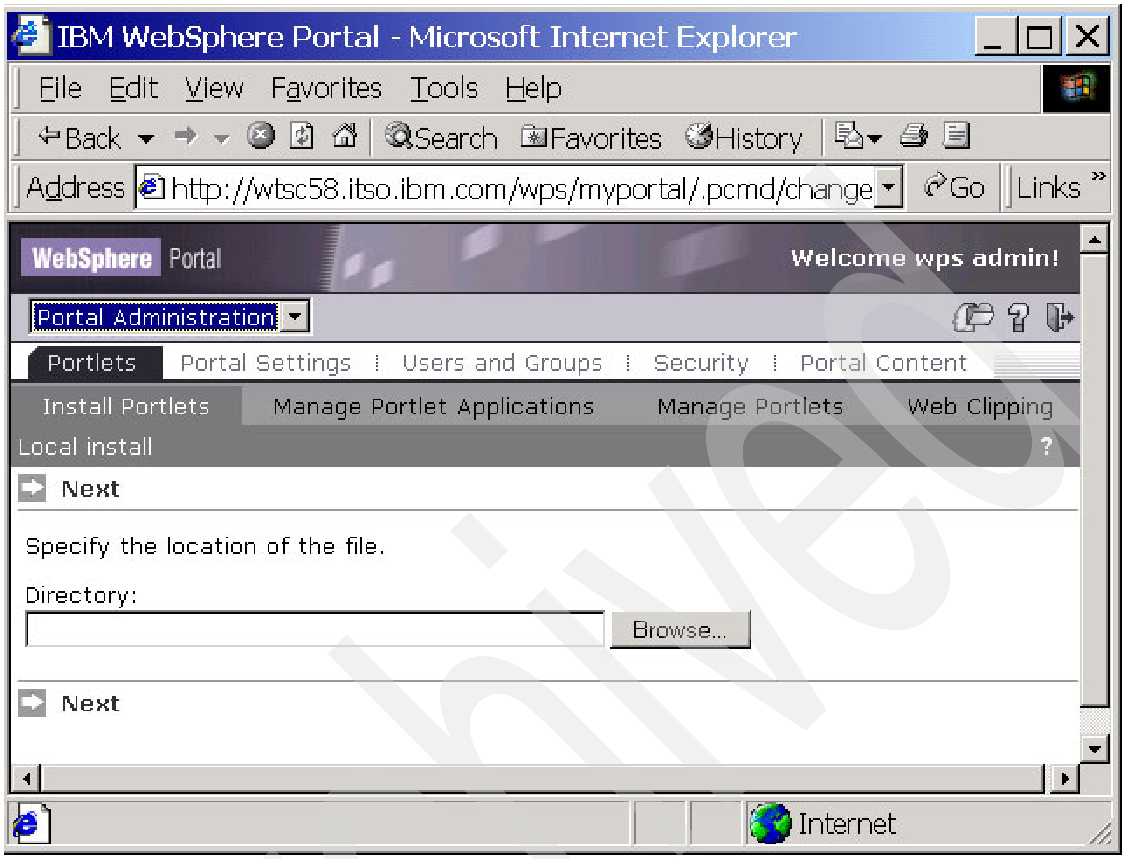
```
http://<HOST_NAME>/wps/portal
```

Note: The default URI of `/wps/portal` may have been changed to another path during installation.

The default page seen in Figure 2-15 on page 61 that you see with four portlets is also seen by all unauthenticated users. After installation, the default configuration permits unauthenticated users access to the top level Places called *Welcome and Work with Pages*.

If you log in as a non-administrative user, you will have access to the Welcome place, Work with Pages, and any other place and pages that all authenticated users have permission to view.

If you log in as `wpsadmin`, the portal administrator, you will have one extra choice in the place menu labeled `Portal Administration`. If you view the Portal Administration option you will see many functions that show up as tabs and sub-tabs. As the portal administrator you can install portlets, manage portlets and portlet applications, determine the portal settings, create/manage users and groups, deal with security and manage portal content. See Figure 4-1 on page 123.



Specify the location of the file.

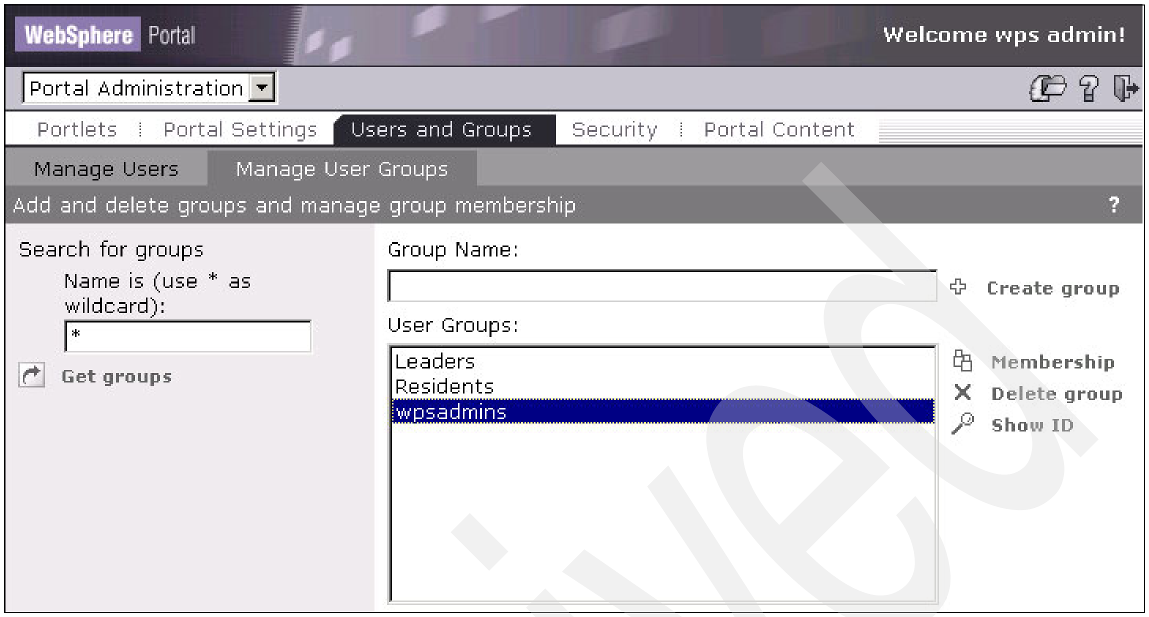
Figure 4-1 Portal Administration screen

4.1.1 Delegating administration

We recommend that you have users who are given some or all administrative authority so that they can act as sub-administrators of either the departmental, geographical, or line of business portal within a large enterprise. Let's see how that can be achieved.

Make sure you are logged in as the portal administrator.

1. Under Portal Administration, navigate down to Users and Groups->Manage User Groups.
2. In the Search for groups area, enter an asterisk for a wildcard and click Get groups. A list of User Groups will be returned as shown in Figure 4-2 on page 124.



Add and delete groups and manage group membership

Search for groups

Name is (use * as wildcard): *

Get groups

Group Name:

Create group

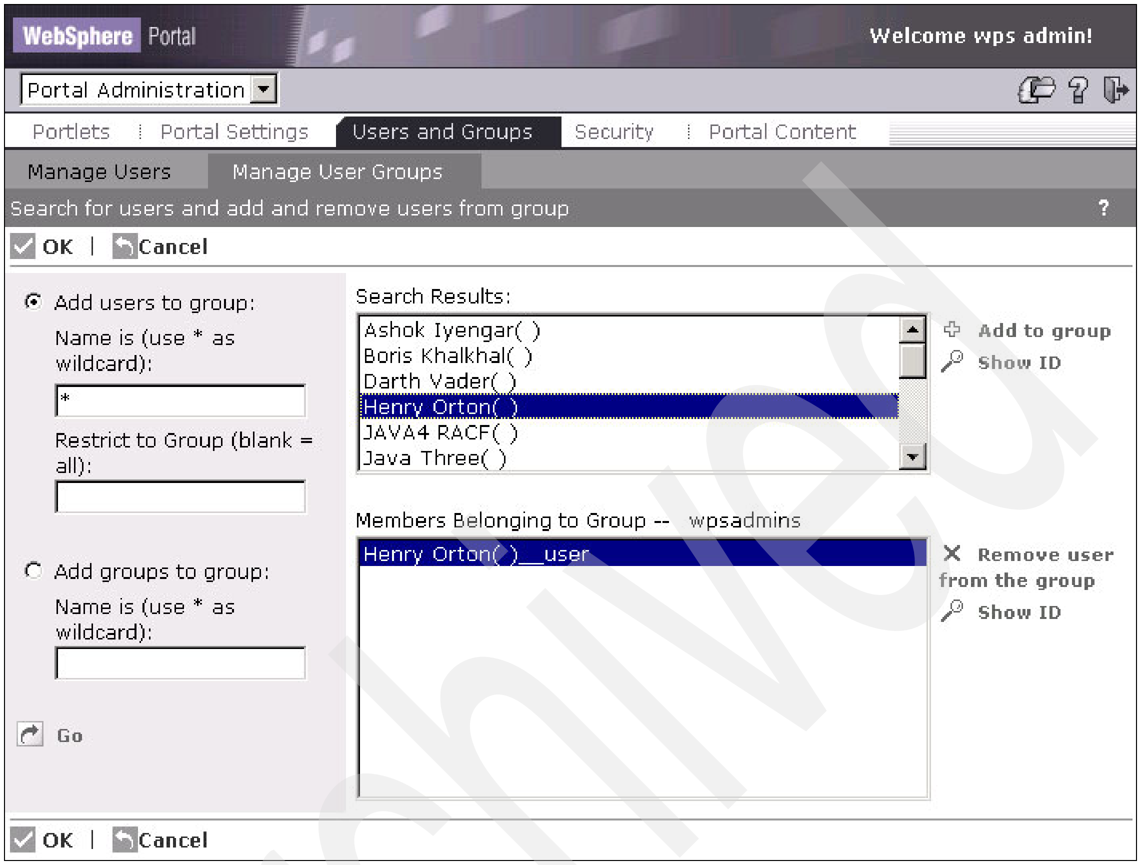
User Groups:

Leaders
Residents
wpsadmins

Membership
Delete group
Show ID

Figure 4-2 Manage User Groups screen

3. Highlight **`wpsadmins`**, and click **`Membership`**.
4. Use the wildcard * to search for all users. Click **`Go`**.
5. From the Search Results, highlight a user and click **`Add to Group`**. That user will show up in the window that reads Members Belonging to Group - wpsadmins. Figure 4-3 on page 125 shows that Henry Orton was added to the `wpsadmins` group.
6. Click OK.



Search for users and add and remove users from group

OK | Cancel

[✓ Add users to group:]

Name is (use * as wildcard): *

Restrict to Group (blank = all):

Add groups to group:

Name is (use * as wildcard):

Go

Search Results:

- Ashok Iyengar( )
- Boris Khalkhal( )
- Darth Vader( )
- Henry Orton( )
- JAVA4 RACF( )
- Java Three( )

Add to group
Show ID

Members Belonging to Group -- wpsadmins

- Henry Orton( )__user

Remove user from the group
Show ID

OK | Cancel

Figure 4-3 Adding a user to wpsadmins group

7. If you log in as that user, in our case, Henry, you will notice that this user has the Portal Administration as one of the places in the place drop down menu.

4.2 Portal customization

WebSphere Portal Version 4.1 introduced the concept of pages that exist within *places*. A drop-down list of places is displayed when a user logs into Portal Server. The places that are displayed depend on the access rights set for the user. Each place can contain one or many pages. Again, users can see only the pages that they have the access rights to view.

A place can be used to organize similar functions or data within portal. For example, a user could have the following places and pages.

- Place: `My hobbies`, with the following pages:
  - `Tennis`, with the following portlets:
    - `Club results`
    - `Wimbledon`
  - `Food and drink`, with the following portlets:
    - `Wine`
    - `Restaurants`
- Place: `News`, with the following pages:
  - BBC, with the following portlets:
    - `Politics`
    - `Sports`
  - Reuters, with the following portlets:
    - `Europe`
    - `USA`
- Place: `Work`, with the following pages:
  - Office, with the following portlets:
    - `email`
    - `Instant messaging`
  - Home, with the following portlets:
    - `email`
    - `synch data`

In this way a hierarchal navigation structure can be built up by the user, or an administrator could use this scheme to create virtual portals within the same Portal Server. Figure 4-4 on page 127 shows a simple drawing of the navigation structure for two Places: the `Portal Administration` place and a user provided `Sports` place.

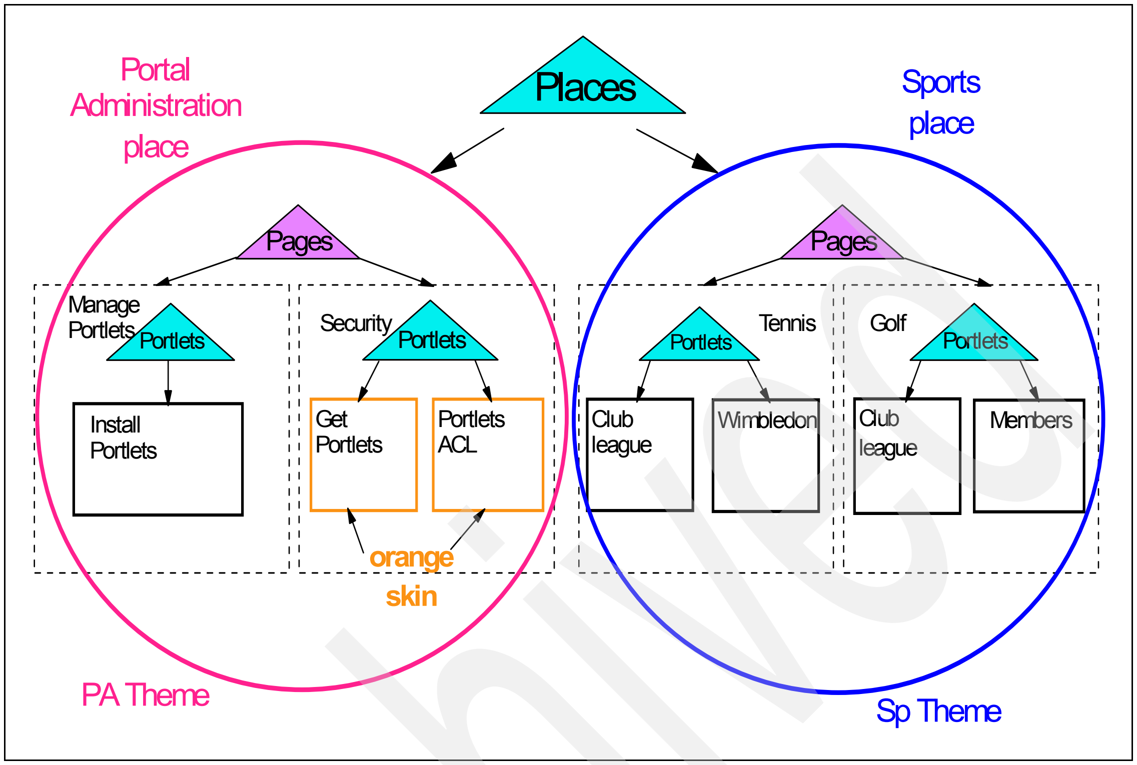


Figure 4-4 Navigation structure using Places, Pages and Portlets

Further customization can be done to the visual appearance of a page, by changing the page's *Theme*, and to the individual appearance of a portlet by changing its *skin*.

In most cases, especially when creating virtual portals, it makes sense to use the same Theme for a place and all of its pages. Therefore, in Figure 4-4 we have circled the Portal Administration place with the color red to represent a single theme applied to all pages in the place, and the Sports place with the color blue to represent a single theme applied to all pages in the place.

In a similar way it often makes sense to use the same portlet skin for all portlets on the same page and also all the portlets within the same place.

The application of Portal Themes and Skins in this way is not enforced, so that the Portal designer or end user can apply variations on this to suit their own branding or visual aid cues as necessary.

4.2.1 Changing themes

WebSphere Portal comes with six different themes — *Admin, Corporate, Engineering, Finance, Science,* and *WebSphere*. The WebSphere Theme is the portal default theme as seen in the Themes and Skins page in Figure 4-5.

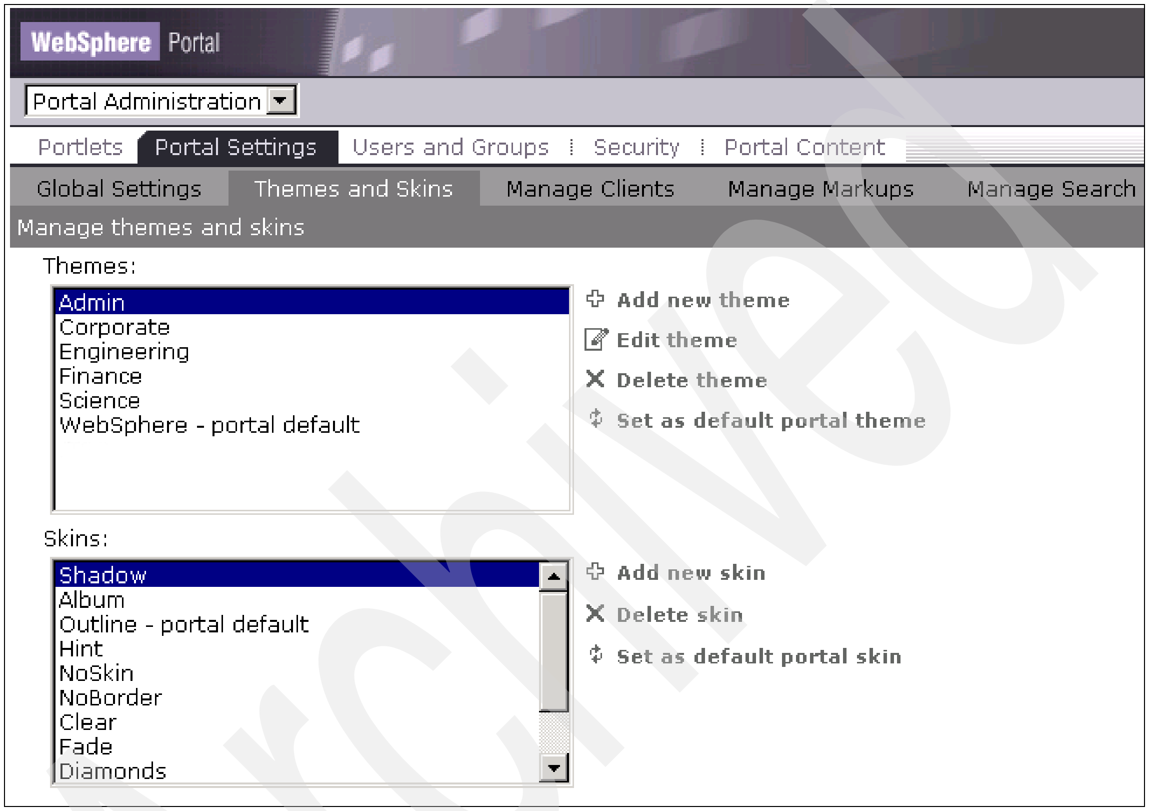


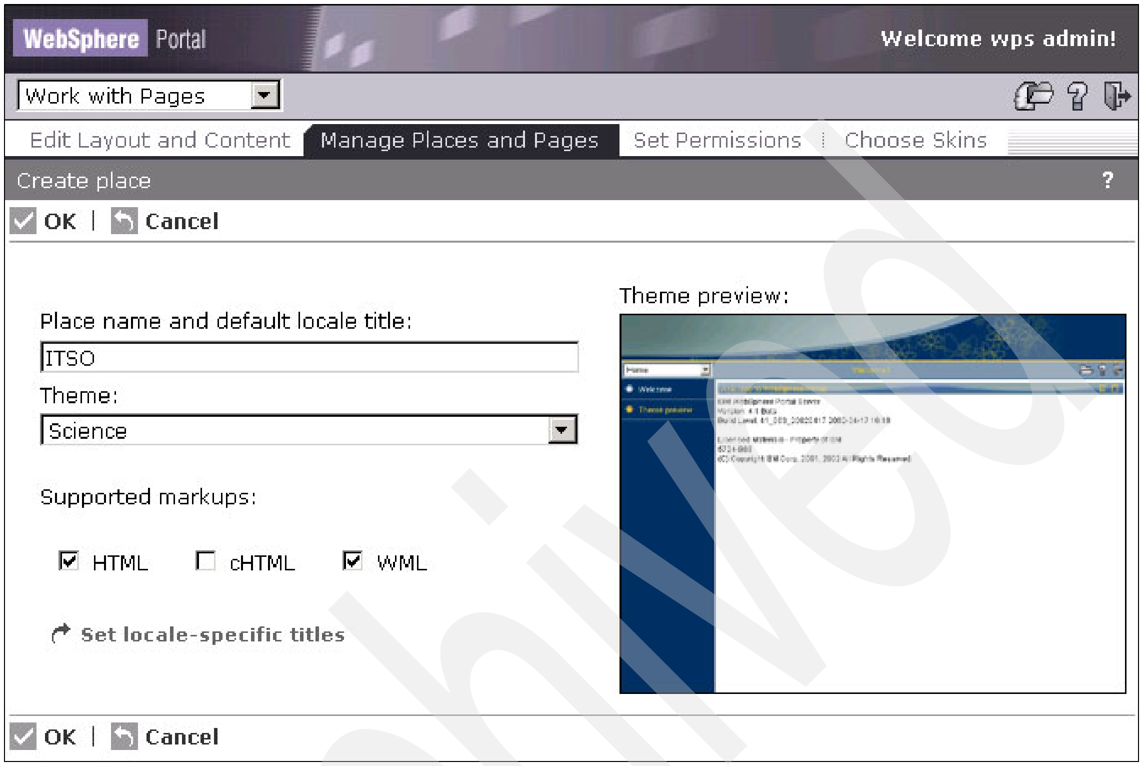
Figure 4-5 Themes and Skins window

The portal administrator has the option to set as default portal theme any one of the themes listed. A better practice is to create a new place and set a particular theme for that place. This way different portal places can have a different look and feel and act as virtual portals.

We used the Science Theme for most of the work done on this redbook project. However, a new place called `ITSO` was created and pages were added to that place as needed. The steps to create this new place were as follows:

1. Navigate to **`Manage Places and Pages`** under **`Work with Pages`**.
2. Enter the Place name. In our case we called our place `ITSO`. Choose **`Science Theme`**. We also chose to support HTML and WML. See Figure 4-6.

3. Clicking **OK** created a place called `ITSO` with the `Science Theme`.



Place name and default locale title: ITSO

Theme: Science

Supported markups: [✓ HTML] cHTML [✓ WML]

Theme preview:

Figure 4-6 Place creation window using the Science Theme

Are we restricted to only these themes? Certainly not. Like everything else in WebSphere Portal themes and skins are customizable. You can create new themes and even add new skins. The easiest thing to do is to start off by copying an existing theme or skin. In the next couple of sections we show you how to create a new skin first, followed by a new theme.

4.3 Adding a new portlet skin

You can create a new skin with the existing installed skins. What we discuss here is creating a new skin, and then using that skin as the default skin in a new theme that we will also create. Because it is easier to copy an existing theme with its skin and modify it, rather than creating a new skin and theme from scratch, this is what we performed and is the recommended practice.

Skins and themes exist for the three markup languages that WebSphere Portal supports — HTML, cHTML, and WML. The examples here only talk about creating a new skin and theme for HTML.

Tip: If you plan to use other markup languages in the portal, remember to add the new theme/skin under sub-directories /WML and /cHTML also.

4.3.1 Create the new skin directory

All skins are found in sub-directories under <WAS_HOME>/apps/WSPORT/WPS Enterprise Application/wps.war/skins directory. Because we want to create a new skin for HTML, go the HTML sub-directory.

1. The skin will be called `Itso`. So we first create a directory named Itso under:

   ```
   <WAS_HOME>/apps/WSPORT/WPS Enterprise Application/wps.war/skins/HTML
   ```

   The directory name is case-sensitive.
2. Copy the contents of the wave skin directory into the newly created Itso directory.
3. There is a `Control.jsp` with a reference to `wave.jpg`. Modify that to read itso.jpg. Save the file.
4. In the directory, you will find two files — `wave.jpg` and `wavepreview.gif`. Rename them to `itso.jpg` and `itsopreview.gif`.
5. Use a graphics editor to modify `itso.jpg` the way you want the new skin to look. Do the same with `itsopreview.gif`. Save the files.

Tip: After creating the new skins directory, and copying files, ensure that the HFS owner, group, and file permissions are updated to be the same as the existing skins directory and files.

4.3.2 Add the skin to WebSphere Portal

Use these steps to add the skin to WebSphere Portal.

1. Make sure you are logged in as the portal administrator. Under **`Portal Administration`**, navigate to **`Portal Settings`**->**`Themes and Skins`**.
2. Click to Add new skin. See Figure 4-7 on page 131.

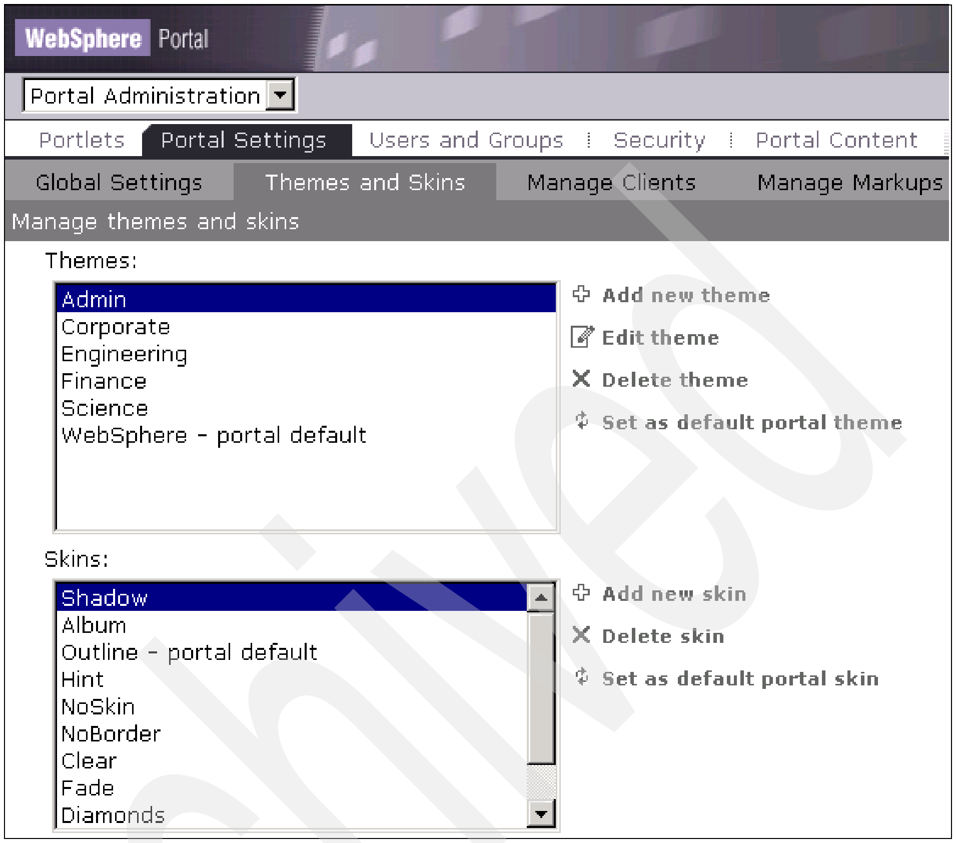
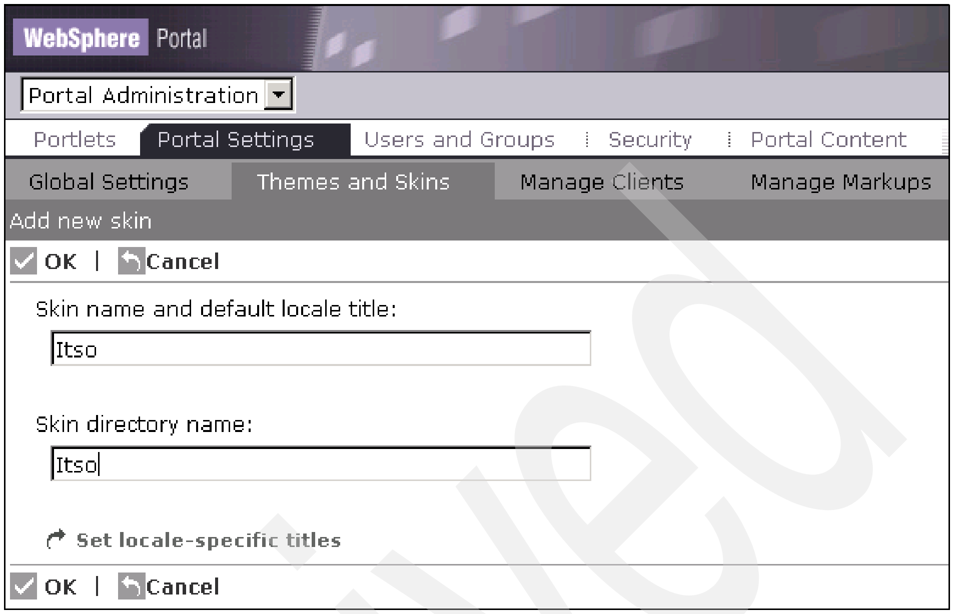


Figure 4-7 Themes and Skins window

3. Enter **`Itso`** for the Skin name.
4. Enter **`Itso`** for the Skin directory name. This name should match the directory name exactly as it was created in the earlier step. See Figure 4-8 on page 132.

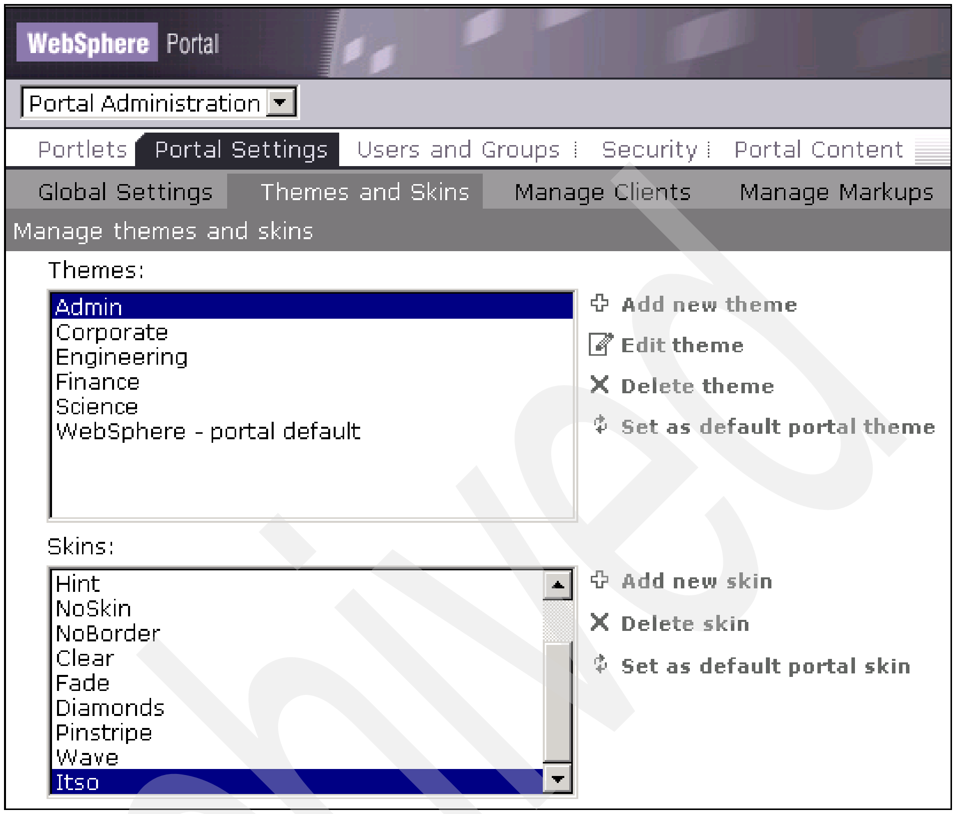


Skin name and default locale title: Itso

Skin directory name: Itso

Figure 4-8 Window to Add a new skin

5. When you click **OK**, the new skin will be registered in WebSphere Portal and will show up at the bottom of the Skins list as seen in Figure 4-9 on page 133.
6. You can now use this new skin within existing themes by adding this skin to the list of skins that the theme already uses.



Manage themes and skins

Themes:

Admin
Corporate
Engineering
Finance
Science
WebSphere - portal default

Skins:

Hint
NoSkin
NoBorder
Clear
Fade
Diamonds
Pinstripe
Wave
Itso

Figure 4-9 Window showing the addition of the new Itso skin

4.4 Adding a new portal theme

All themes are found in sub-directories under the directory:

```
<WAS_HOME>/apps/WSPORT/WPS Enterprise Application/wps.war/themes
```

Because we want to create a new theme for HTML, go the HTML sub-directory. Typically the theme exists in a directory with the same name.

4.4.1 Create the new theme directory

Use these steps to create the new theme directory:

1. The theme will be called `ITSO`. So we first create a directory name ITSO under:

   ```
   <WAS_HOME>/apps/WSPORT/WPS Enterprise Application/wps.war/themes/HTML
   ```

The directory name is case-sensitive.

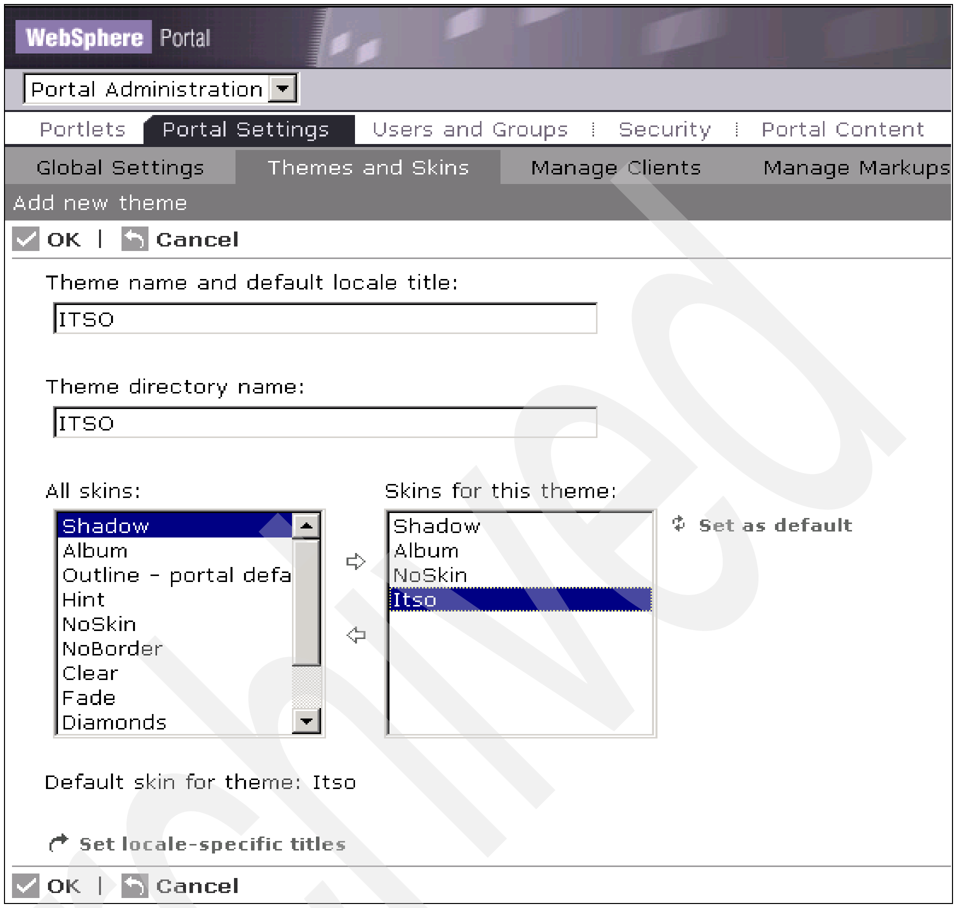
2. Copy the contents of the Science Theme directory into the newly created ITSO directory.
3. Edit the style sheet file, `Styles.css` to display things the way you want when this new theme is displayed. Save the file.
4. Use a graphics editor to modify `banner.jpg` the way you want the banner in the new theme to look. Save the file.

Tip: After creating the new theme directory, and copying files, ensure that the HFS owner, group, and file permissions are updated to be the same as the existing theme directory and files.

4.4.2 Add the theme to WebSphere Portal

Use these steps to Add the theme to WebSphere Portal:

1. Make sure you are logged in as the portal administrator. Under **`Portal Administration`**, navigate to **`Portal Settings`**->**`Themes and Skins`**.
2. Click to Add new Theme. See Figure 4-7 on page 131.
3. Enter **`ITSO`** for the Theme name.
4. Enter **`ITSO`** for the Theme directory name. This name should match the directory name exactly as it was created in the earlier step. See Figure 4-10 on page 135.
5. Choose **`Shadow`**, **`Album`**, **`NoSkin`**, and **`Itso`** as the skins that make up the new theme.
6. Highlight **`Itso`** in the **`Skins`** for this theme list and set it as the default by clicking the **`Set as default link`**.
7. Click **`OK`**. The newly-created ITSO theme should be ready for use.



Add new theme

Theme name and default locale title: ITSO

Theme directory name: ITSO

All skins: Shadow, Album, Outline - portal defa, Hint, NoSkin, NoBorder, Clear, Fade, Diamonds

Skins for this theme: Shadow, Album, NoSkin, Itso

Default skin for theme: Itso

Figure 4-10 Screen to add a new theme

The newly created ITSO Theme and the Itso Skin should be available for use just like any other out-of-box theme and skin. Our intent here was to show you the directory where the WebSphere Portal themes and skins files are found on the z/OS and OS/390 platform and the files that need to be modified to create new themes and skins. It is not at the same location as found on the distributed platform, but the mechanics of creating or adding a new theme and skin are the same.

To illustrate what was done here, a new place called myITSO was created that used the ITSO theme. And since the Itso skin was set as default, the Reminder portlet that was placed on the myResidency page assumes the new Itso skin. See Figure 4-11 on page 136.

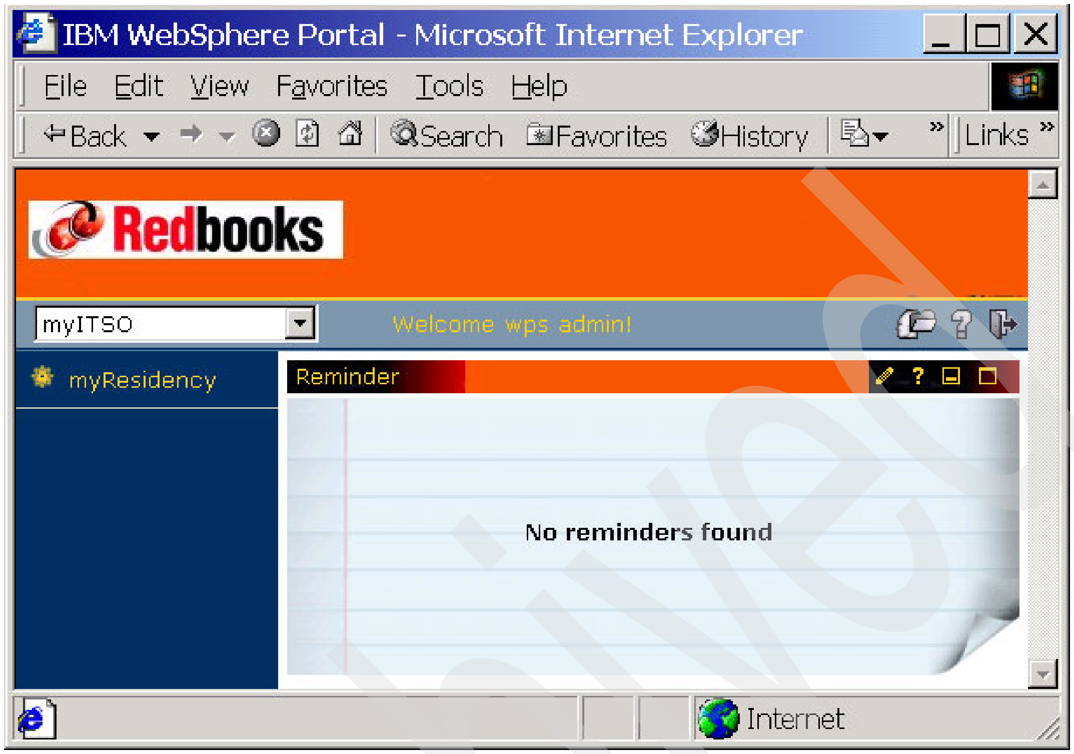

IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History Links
Redbooks
myITSO
Welcome wps admin!
myResidency
Reminder
No reminders found
Internet

Figure 4-11 Screen shot of a portal page using the new ITSO theme and Itso skin

4.5 Log Files

Another area that is different from the distributed platform is the location where the portal server log files are stored. On the z/OS and OS/390 platform the portal log files can be found in <WPS_HOME>/log. On our system that was /PortalServer/log, and the log files are named wps_<yyyy.mm.dd-hh.mm.ss>.log.

If you develop your own portlets, they are able to write to this log by using the PortletLog object from the Portlet API (Example 4-1 on page 137).

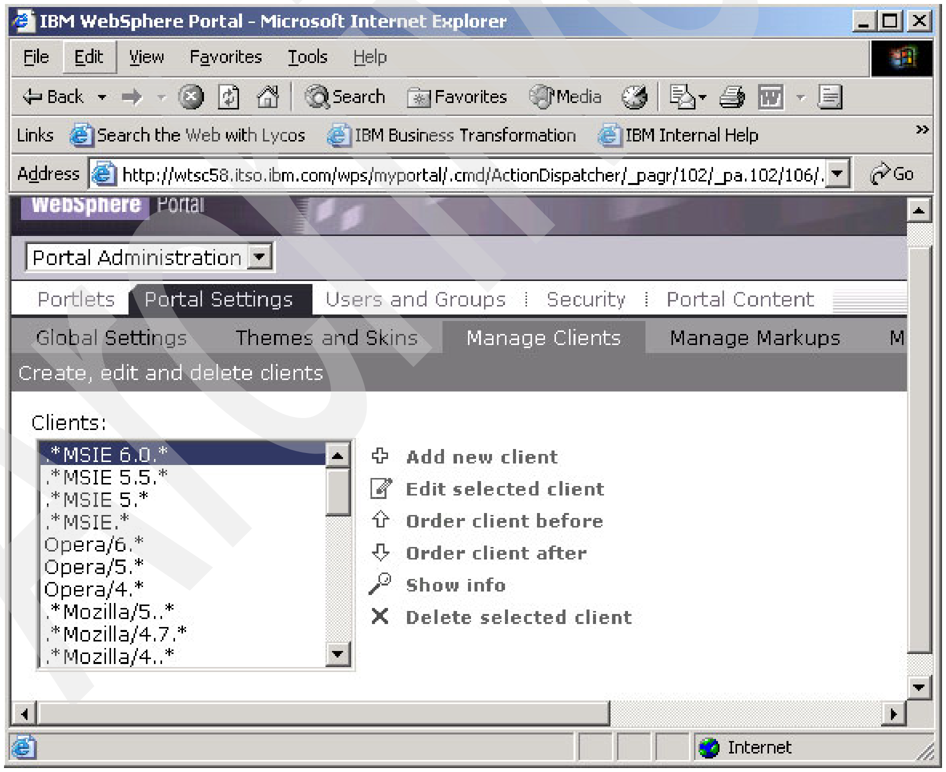
Example 4-1 Using the Portal log from a portlet

```
PortletLog log = getPortletConfig().getContext().getLog();

if (log isDebugEnabled()) log.debug("Debug enabled:" + debugMsg);
if (log isWarnEnabled()) log.debug("Warnings enabled:" + warningMsg);
if (log isInfoEnabled()) log.debug("Info enabled:" + infoMsg);
if (log isErrorEnabled()) log.debug("Error enabled:" + errorMsg);
```

It is also possible to use typical Java console statements such as `System.out.println("My debug statement");` which will show up in the SYSPRINT (JES2 JOB LOG) for the portal server region.

4.6 Device support

When installed WebSphere Portal supports a variety of client device types, for example browsers from Microsoft, Opera and Mozilla, as shown in Figure 4-12.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address
Go
WebSphere Portal
Portal Administration
Portlets Portal Settings Users and Groups Security Portal Content
Global Settings Themes and Skins Manage Clients Manage Markups M
Internet

Address: http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispatcher/_pagr/102/_pa.102/106/.

Create, edit and delete clients

Clients:

.*MSIE 6.0.*
.*MSIE 5.5.*
.*MSIE 5.*
.*MSIE.*
Opera/6.*
Opera/5.*
Opera/4.*
.*Mozilla/5..*
.*Mozilla/4.7.*
.*Mozilla/4..*

Add new client
Edit selected client
Order client before
Order client after
Show info
Delete selected client

Figure 4-12 WebSphere Portal supported clients

The list of supported clients includes devices that typically use mark-up languages HTML, WML and cHTML. Because WebSphere Portal is able to detect the type of client it can render the response using the appropriate mark-up and in case of cHTML and WML this means support for many popular types of mobile or cell phones.

We tested the device support for WML using `the M3Gate` emulator from *Numeric Algorithm Laboratories*. After using M3gate to access the portal home page, we received the display shown in Figure 4-13 on page 138. Note that because of the device capability, only text links are rendered for display by WebSphere Portal, using WML.

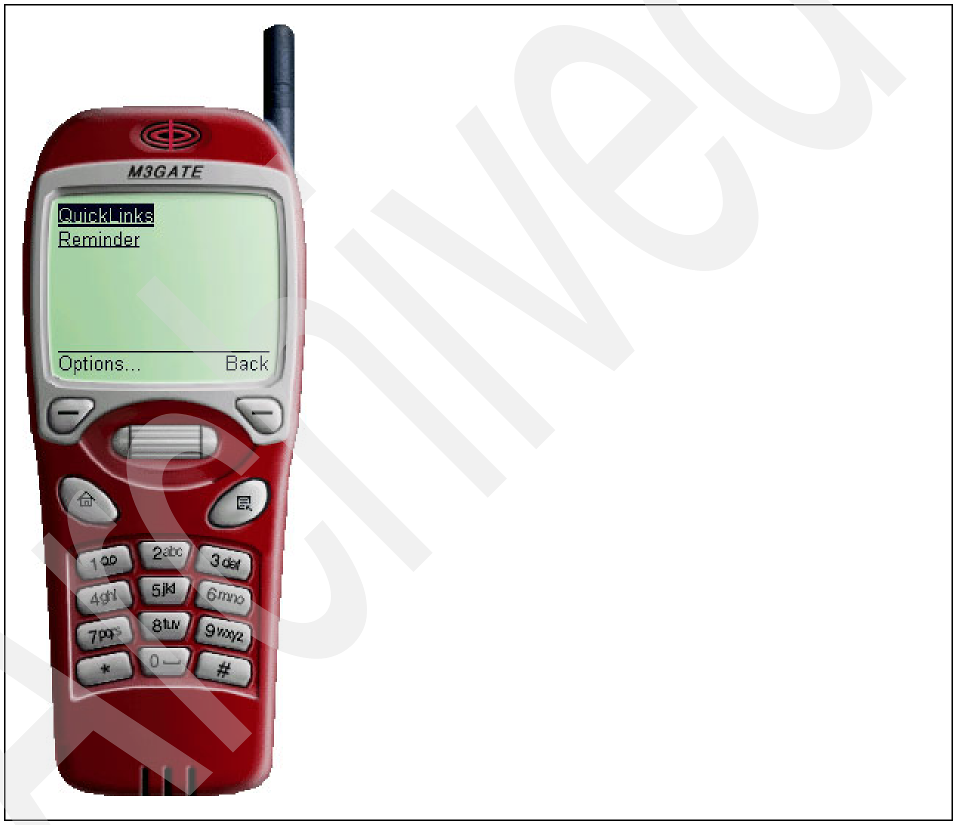


Figure 4-13 Portal home page on a WML device

Using M3Gate, we clicked on **`Options`** and navigated to the `ITSO` place as shown in Figure 4-14 on page 139. The ITSO place also has the `HelloWorld` home page visible from a browser client, but this was not shown in the M3Gate view of the place.

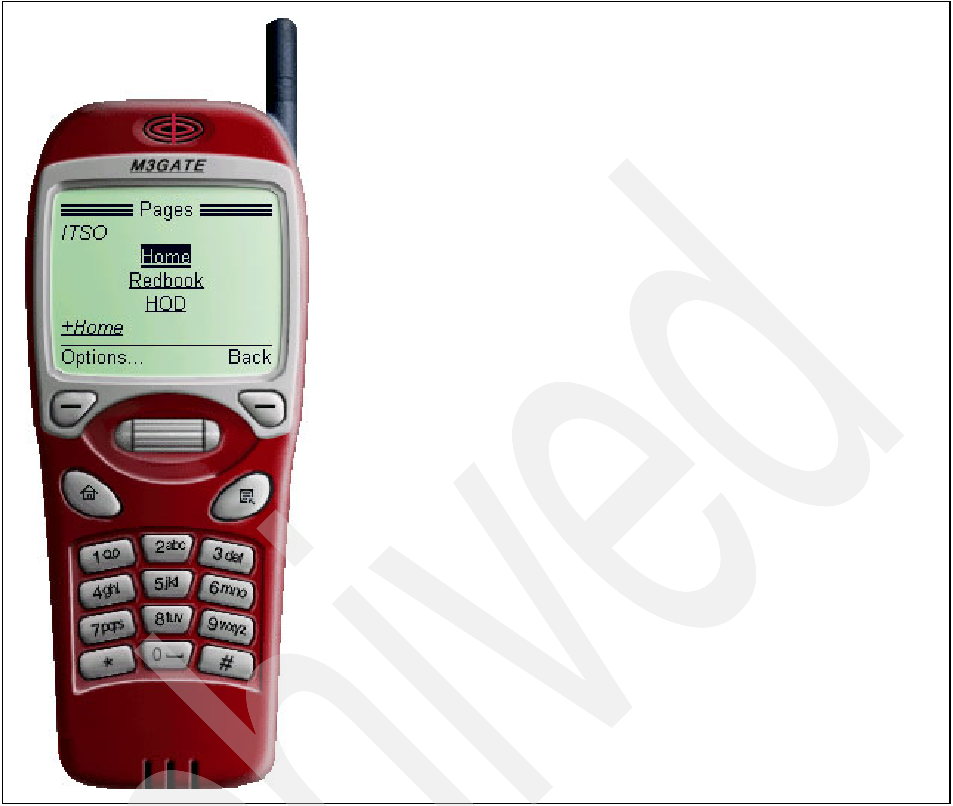
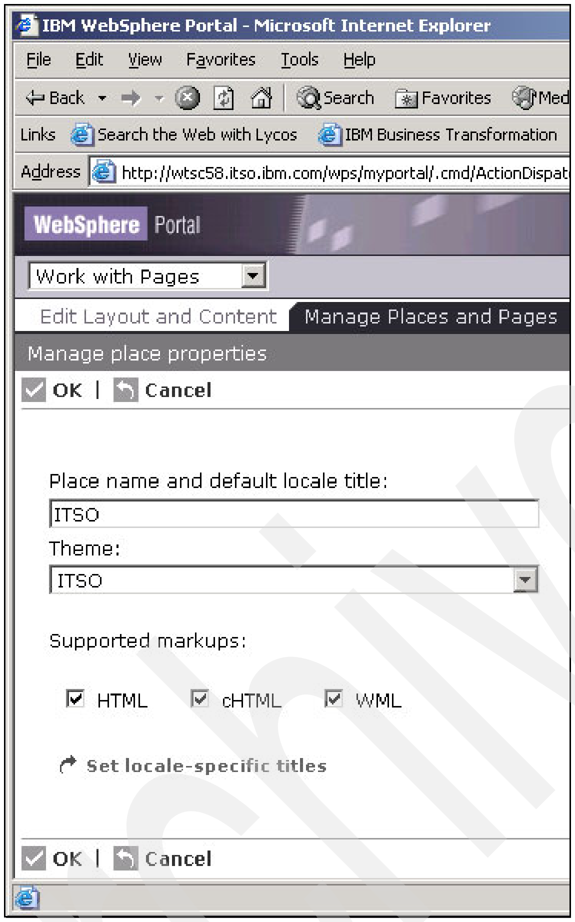


Figure 4-14 Logged in page, ITSO place, WML device, Hello World place not visible

To rectify this we used a normal browser, logged in as the portal administrator, navigated to **`Manage Places and Pages`**, selected the **`ITSO`** theme and checked the **`cHTML`** and **`WML`** options as shown in Figure 4-15 on page 140.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Med
Links Search the Web with Lycos IBM Business Transformation
Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispat
WebSphere Portal
Work with Pages
Edit Layout and Content
Manage Places and Pages
OK | Cancel
Set locale-specific titles
OK | Cancel

Manage place properties

Place name and default locale title: ITSO

Theme: ITSO

Supported markups:

[✓ HTML] [✓ cHTML] [✓ WML]

Figure 4-15 Enabling WML for the ITSO place

Once again using the M3Gate mobile/cell phone emulator we navigated to the **`ITSO`** place and this time were rewarded by a link to the **`Hello World`** page as seen in Figure 4-16 on page 141.

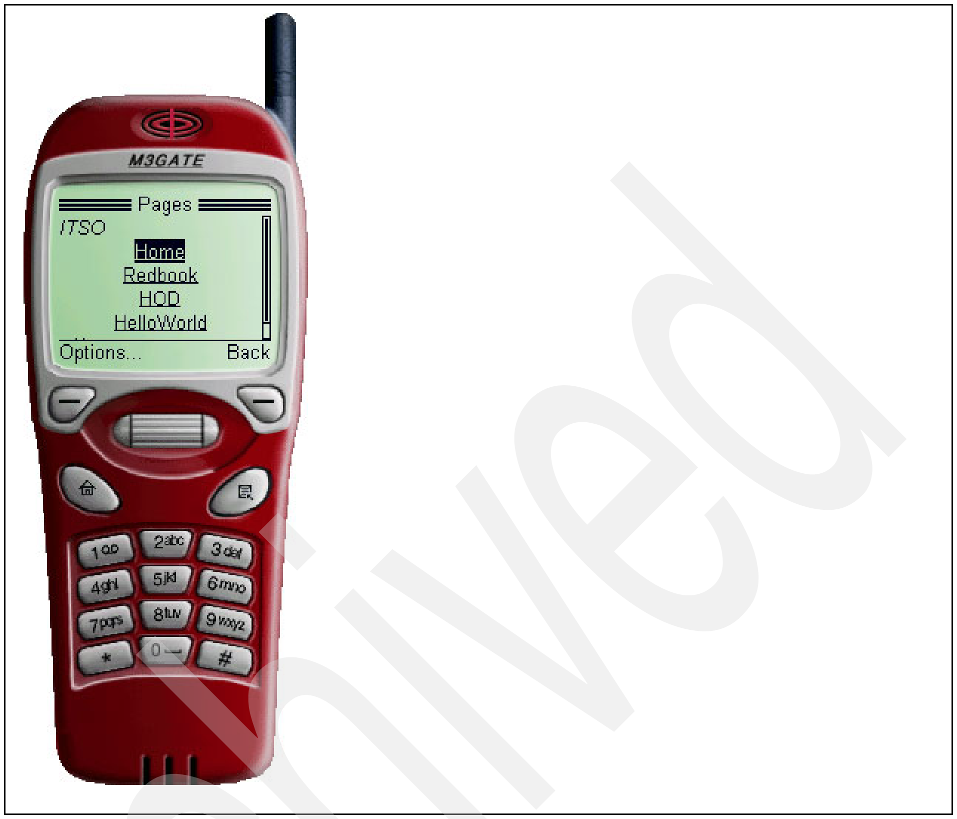


Figure 4-16 Hello World in the ITSO place visible

4.7 Managing Portlets in z/OS

Section 6.3, “Portlet deployment” on page 194 describes how to add and deploy new portlets. After installation of WebSphere Portal on z/OS, a number of portlets are deployed and activated during the portlet deployment post-installation step, as described in “Step twelve: Deploy the base portlets” on page 58.

Portlet WAR files can package multiple portlets, and these are called *Portlet Applications*. The administrative browser GUI allows you to manage individual portlets, or portlet applications containing multiple portlets. As previously noted in 6.3, “Portlet deployment” on page 194, the deployment process is disruptive to the Portal Server, and if Portal Server is running at the time of deployment it will be stopped and restarted for the deployment of each portlet found within a portlet application. Therefore, we recommend stopping Portal Server manually before making portlet application changes that affect many portlets.

4.7.1 Updating portlets

In a similar way to portlet deployment, updating an existing portlet with a new WAR file is also a two step process, first performed by the administrator from the `Portal Administration` place, pages **`Manage Portlet Applications`** or **`Manage Portlets`**, and subsequently from TSO or USS to submit a job that runs script `WPAConfig`. This also applies to deleting a deployed portlet or portlet application.

Important:

- On the version of WebSphere Portal for z/OS that we tested, based on PTF2, when performing an update to a portlet that is already installed you should use the same WAR filename as the one that is already deployed.
- The second step, running WPACONF job or WPAConfig script, used for deploying, updating and deleting portlets, will cause the portal server to be stopped and restarted for each portlet, if it was already running. Therefore, we recommend that you stop the portal server before running this job as the second deployment step for portlets.

4.7.2 Installed but not deployed portlets

A number of portlets are provided with WebSphere Portal for z/OS, but not deployed during the post-installation steps. This set of available and deployable portlets is in directory <WPS_ROOT>/PortalServer/app/DeployablePortlets. To deploy them, the WAR files should be transferred to a workstation from where an administrator can use the browser administrative interface to perform Step 1 of the deployment process, followed by deploy Step 2 from the z/OS system.

4.8 Managing the portal environment

In this section we provide some information about areas that we found important for consideration when managing WebSphere Portal Server on z/OS. Some of these areas have already been covered in previous sections or other chapters, but they are collected together here for reference:

1. Portlet deployment is disruptive to the server as discussed in 6.1, “Portlet development” on page 182 and 4.7, “Managing Portlets in z/OS” on page 141.
2. Portlet deployment uses the Systems Management API (SMAPI) which can have unforeseen side-effects with administrators also using SMEUI to perform updates to the Application Server.

For example lets say one systems administrator is using SMEUI to modify either the same portal server region, or another J2EE application running in a different server region on the same application server. To do this they will have created a new conversation using SMEUI, made some configuration changes, **`validated`** and **`committed`** the conversation.

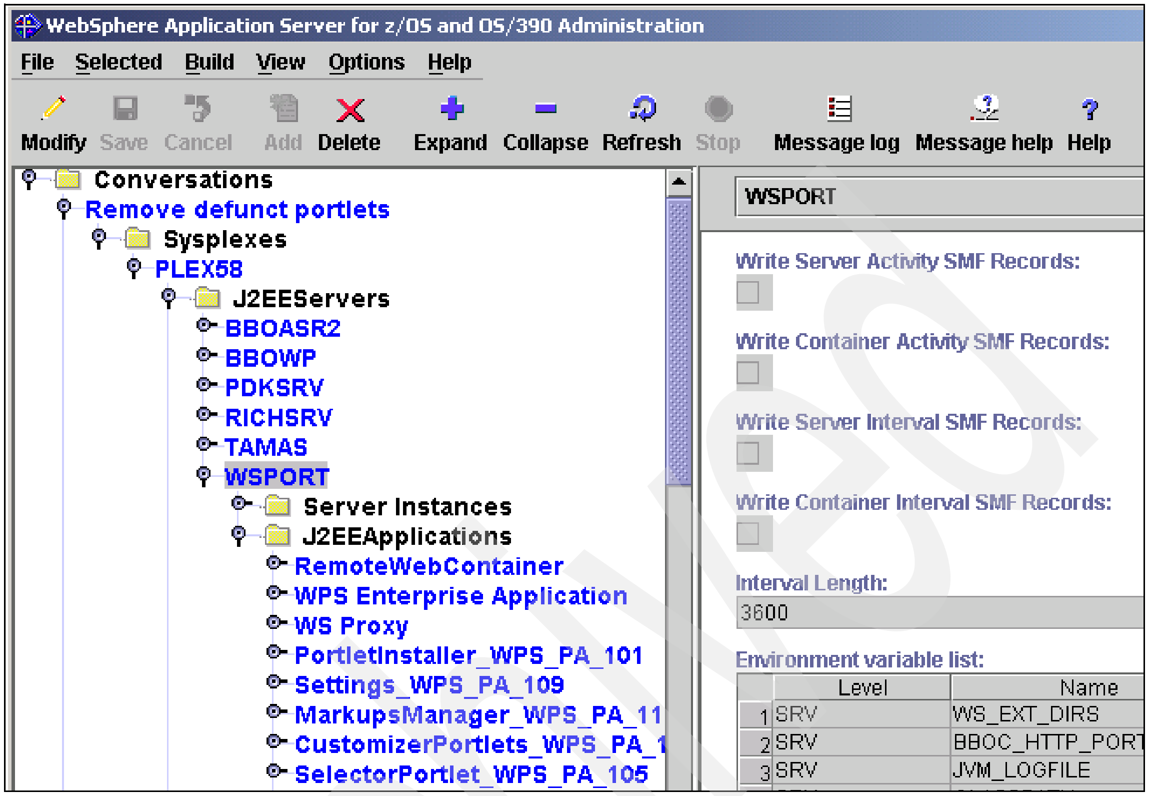
If at the same time this has been happening, another systems administrator deploys a portlet, and since the deployment process also uses SMAPI it is possible for the deployment process to create a new conversation after the creation of the conversation done by the administrator using SMEUI.

The unfortunate side-effect of this will be that the administrator using SMEUI will not be able to activate their conversation and configuration changes, because the conversation that the changes were modelled on have been replaced by the conversation used to deploy the portlet(s).

3. A deployed portlet is installed as a J2EE application (EAR file) running on the Portal Server. Therefore, environment variables for the J2EE server and modifications to the installed EAR files can be done from the SMEUI Administrative GUI (Figure 4-17 on page 144).

   SMEUI can be used to define new environment variables to be used by new portlets that require additional classes/jars to be added to the J2EE container.

   However, we do not recommend that administrators use the SMEUI to manage individual J2EE Portlet Applications, because this is normally managed via the deployment process. In the case of an error however, for example, updating a portlet using a different WAR file name, which, as described in 4.7.1, “Updating portlets” on page 142, is not supported, then the orphan or defunct EARs can be deleted using SMEUI. Figure 4-17 on page 144 shows SMEUI Administration with a list of installed J2EE Portlet applications and the J2EE server environment variables.


WebSphere Application Server for z/OS and OS/390 Administration
File Selected Build View Options Help
Modify Save Cancel Add Delete Expand Collapse Refresh Stop Message log Message help Help

```
Conversations
  Remove defunct portlets
    Sysplexes
      PLEX58
        J2EEServers
          BBOASR2
          BBOWP
          PDKSRV
          RICHSRV
          TAMAS
          WSPORT
            Server Instances
            J2EEApplications
              RemoteWebContainer
              WPS Enterprise Application
              WS Proxy
              PortletInstaller_WPS_PA_101
              Settings_WPS_PA_109
              MarkupsManager_WPS_PA_11
              CustomizerPortlets_WPS_PA_1
              SelectorPortlet_WPS_PA_105
```

WSPORT

Write Server Activity SMF Records: ☐

Write Container Activity SMF Records: ☐

Write Server Interval SMF Records: ☐

Write Container Interval SMF Records: ☐

Interval Length: 3600

Environment variable list:

	Level	Name
1	SRV	WS_EXT_DIRS
2	SRV	BBOC_HTTP_PORT
3	SRV	JVM_LOGFILE

Figure 4-17 SMEUI Administrative functions for Portal J2EE applications

4. The Portal log needs to be managed.

   As noted in 4.5, “Log Files” on page 136 the portal log is a series of time stamped files created in a directory in the HFS. These log files are not automatically deleted after a period of time, so administrative procedures need to be in place to manage their growth.

5. Portal configuration files are maintained in ASCII.

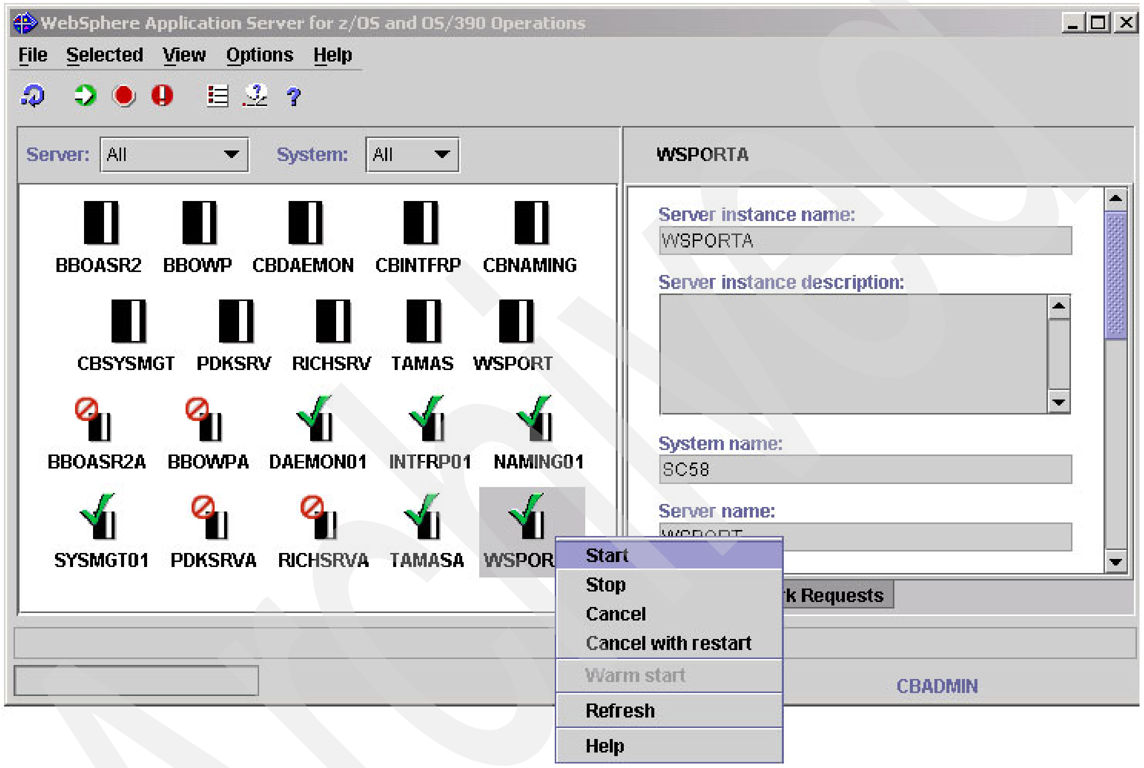
   Many of the portal configuration files, for example, files located in <WPS_PATH>/libapp/config are in ASCII and are not viewable or editable from TSO ISH or USS. IBM provides a number of free z/OS tools for download, including an ASCII editor available at:

   http://www-1.ibm.com/servers/eserver/zseries/zos/unix/bpxa1ty2.html

6. Need to manage file attributes, owner, group, permissions, if using USS.

   Some operations in the portal require the creation and update of directories and files in the HFS. The directory file permissions, including owner and group need to be preserved across manual changes or some portal functions will not work correctly.

7. Administrative and Operations control of the Portal Server, for J2EE configuration and start/stop of the server region, can be controlled in the usual way for any WebSphere Application Server z/OS application by using the SMEUI tool from a workstation. Figure 4-18 shows an example of using SMEUI Operations to start the Portal server region on our system.



Server: All
System: All

WSPORTA

Server instance name: WSPORTA
Server instance description:
System name: SC58
Server name: WSPORT

Figure 4-18 Using SMEUI Operations to start the Portal server

5

WebSphere Portal in a Sysplex

This chapter was written by the WebSphere Portal development team to show some aspects of the Portal in a sysplex environment. Sysplex is a unique feature of the zSeries architecture and we thought it might be a good idea to show the sysplex scaling and availability possibilities when using Portal on zSeries.

5.1 Configuring WebSphere Portal in a Sysplex

Configuring WebSphere Portal in a sysplex environment means that you set up the application server of WebSphere Portal in WebSphere Application Server with replicated server instances on multiple systems within a sysplex. Using a sysplex distributor in front of the portal server instances makes this change completely transparent to the client. That is, from a client perspective, the application server of WebSphere Portal listens to a single Virtual IP Address (VIPA) that is defined to the sysplex distributor. The sysplex distributor then routes incoming requests to the single server instances of the application server, depending on availability and workload on the hosting system.

Therefore, the benefits of such a configuration are:

- Increased performance
- Workload balancing between the systems within the Sysplex
- High availability

For further information, refer to *z/OS Communications Server IP Configuration Guide,* SC31-8775-01*.*

5.2 Setting up the infrastructure

In the configuration sample here, the following infrastructure is assumed: an existing parallel sysplex with coupling facility, shared HFS, WLM running in goal mode, and DB2 Version 7 running in datasharing mode.

5.2.1 Sysplex distributor

To set up a sysplex distributor, you need to enable an XCF network, define one or more virtual IP addresses, and a routing protocol, such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF). The routing protocol is an essential element of this setup, because availability and routes of the IP address can change dynamically. Therefore, it is necessary to propagate routes dynamically using a routing protocol. In the sample configuration, OSPF is chosen to manage routing within the sysplex.

A sysplex distributor defines a virtual IP address (VIPA) that represents a sysplex-wide IP address known to all systems in a sysplex. Therefore, such an address is also called a cluster address. The VIPA distributes IP requests from the distributing IP stack to other IP stacks (including its own) through the internal XCF network of the sysplex. Distribution is done depending on workload within the sysplex.

In the sample configuration, the systems WPS3 and WPS4 are from an existing sysplex. System WPS3 owns the distributing IP stack and WPS4 is defined as the VIPA backup.

Three VIPAs are defined to allow for more than four ports to be distributed within the sysplex. This is necessary because the TCP/IP APAR (PQ65205), which allows the distribution of more than four ports per VIPA, is not applied.

5.2.2 Cluster addresses

The following IP address is associated with the DNS name:

```
9.152.88.233 = wpsplex1.boeblingen.de.ibm.com
```

This address distributes incoming requests to the WebSphere Application Server LDAP (port: 1389), the WebSphere Application Server daemon (port: 5555 and 5556), and the System Management of WebSphere Application Server (port: 9000).

```
9.152.88.234 = wpsplex2.boeblingen.de.ibm.com
```

This second address distributes requests to WebSphere Portal (port: 8082), the HTTP server (port: 80), the sample application of WebSphere Application Server (port: 8081), and the WebSphere Portal LDAP (port: 2389).

```
9.152.88.235 = wpsplex3.boeblingen.de.ibm.com
```

This third address distributes requests for TELNET (port: 23), OTELNET (port: 623), and FTP (port: 21).

5.2.3 XCF addresses

XCF addresses are the IP addresses of the single systems in the internal XCF network of the sysplex. XCF for TCP/IP is implemented through VTAM-to-VTAM connections. It is enabled in z/OS and VTAM® either by specifying startup parameters `XCFINIT=YES or HPR=RTP`, or by activating the VTAM major node ISTLSXCF. For configuration details, refer to *z/OS Communications Server: SNA Network Implementation,* SC31-8777.

The XCF devices for TCP/IP (EZAXCFS4, and so on) are dynamically generated by specifying `IPCONFIG DYNAMICXCF` in the profile `TCP.PROFILE`. The sysplex distributor then distributes incoming client requests to the XCF addresses depending on availability and workload, or by complying with policies specified for the policy agent of the requested services.

The XCF addresses are:

`9.152.88.35` for system WPS3

`9.152.88.36` for system WPS4

5.2.4 Static IP addresses

Static IP addresses are the IP addresses of devices that are physically connected to the intranet or Internet.

The static IP addresses for WPS3 are:

Address	Device	IP name DNS)
9.152.87.143	LINKETH	**boewps3.boeblingen.de.ibm.com**
9.152.64.117	LINKVIA	none

The static IP addresses for WPS4 are:

Address	Device	IP name (DNS)
9.152.87.144	LINKETH	**boewps4.boeblingen.de.ibm.com**
9.152.64.118	LINKVIA	none

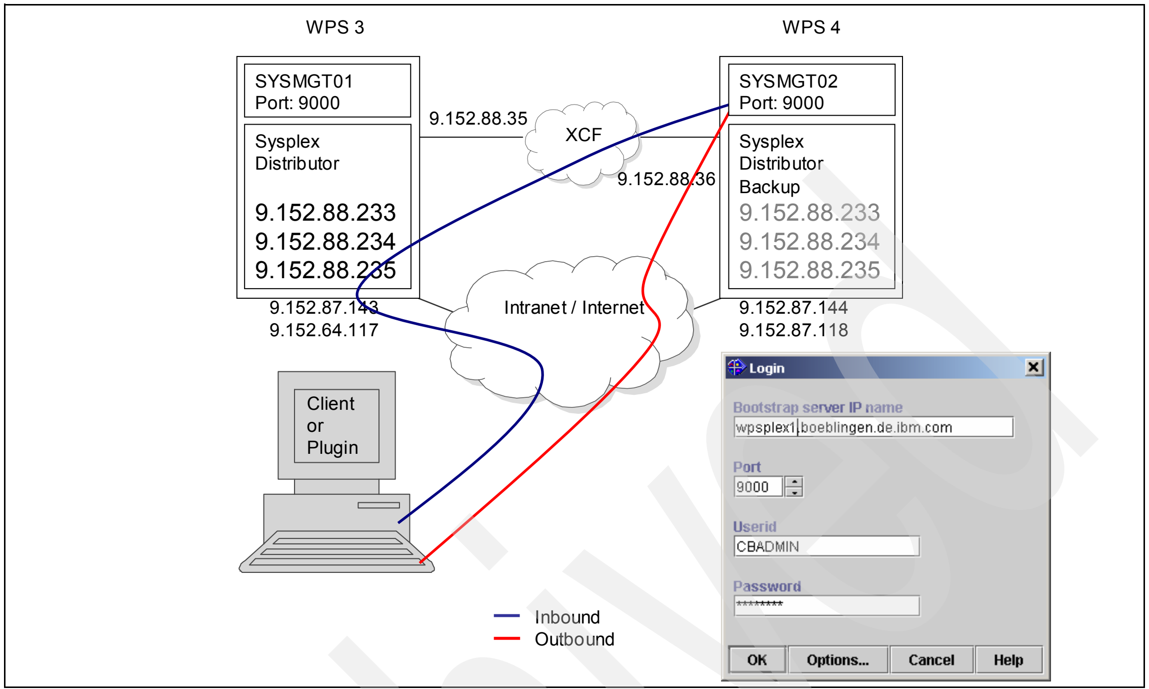
LINKETH is the device that was previously defined for the system WPS3 and WPS4, respectively.

LINKVIA is the device that was added per stack, because device LINKETH is not connected to a subnet that provides OSPF as the routing protocol.

In a real, high availability solution including sysplex distributor and WebSphere Application Server, the IP name of the static IP address should be defined on the OSPF-managed device, or should be a static VIPA. This is not the case in the test environment.

If, for example, device LINKETH is not accessible in the test environment, then IP names **boewps3** and **boewps4** are not available. This is because in the test environment, LINKETH and LINVIA are fast Ethernet devices defined through QDIO physically sharing the same device. Therefore, the test environment demonstrates the principles and is not an actual high availability solution.

Figure 5-1 on page 151 shows the principles of a sysplex distributor.



Bootstrap server IP name: wpsplex1.boeblingen.de.ibm.com
Port: 9000
Userid: CBADMIN
Password: ********

Figure 5-1 Principles of a sysplex distributor

That is, for the client, the System Management of WebSphere Application Server is accessible as if it was a single server, under a single cluster address (`wpsplex1.boeblingen.de.ibm.com`). In this figure, a client, for example, the System Management End User Interface (SMEUI) of WebSphere Application Server, connects through the sysplex distributor to the System Management of WebSphere Application Server. The client request is distributed to WPS4, depending on the workload and availability. For the client, this processing is transparent.

Implementation of the sysplex environment

As a first step, the IP configuration profiles for the systems are updated. Example 5-1 and Example 5-2 are sample profiles for the test environment.

Example 5-1 TCP.PROFILE of system WPS3

```
IPCONFIG SYSPLEXROUTING DATAGRAMFWD
DYNAMICXCF 9.152.88.35 255.255.255.240 1

VIPADYNAMIC
VIPADEFINE MOVE IMMEDIATE 255.255.255.248 9.152.88.233 9.152.88.234
VIPADEFINE MOVE IMMEDIATE 255.255.255.248 9.152.88.235
VIPADISTRIBUTE DEFINE 9.152.88.233 PORT 1389 5555 5556 9000
```

```
 DESTIP 9.152.88.35 9.152.88.36
VIPADISTRIBUTE DEFINE 9.152.88.234 PORT 80 2389 8081 8082
 DESTIP 9.152.88.35 9.152.88.36
VIPADISTRIBUTE DEFINE 9.152.88.235 PORT 21 23 623
 DESTIP 9.152.88.35 9.152.88.36
ENDVIPADYNAMIC
.
.
HOME
   9.152.87.143 LINKETH
   9.152.64.117 LINKVIA
```

Example 5-2   TCP.PROFILE of system WPS4

```
IPCONFIG SYSPLEXROUTING DATAGRAMFWD
DYNAMICXCF 9.152.88.36 255.255.255.240 1
;
VIPADYNAMIC
VIPAbackup 100 9.152.88.233
VIPAbackup 100 9.152.88.234
VIPAbackup 100 9.152.88.235
ENDVIPADYNAMIC
.
.
HOME
9.152.87.144 LINKETH
9.152.64.118 LINKVIA
```

Next, OSPF routing for the sysplex IP addresses is set up. This configuration depends on your network topology, and is required for the sysplex distributor to manage dynamic IP addresses.

Example 5-3 shows the OSPF configuration for the system that owns the sysplex distributor, *WPS3.*

Example 5-3   OSPF definition of system WPS3

```
RouterID=9.152.64.117;
Area
     Area_Number=0.0.0.3
     stub_Area=YES
     Authentication_Type=Password;
OSPF_Interface IP_Address=9.152.64.117
     Name=LINKVIA
     Subnet_mask=255.255.255.240
     MTU=1492
     Attaches_To_Area=0.0.0.3
     Hello_Interval=1
     Dead_Router_Interval=4
     Authentication_Key="xx";
OSPF_Interface IP_Address=9.152.88.*
     Name=XCFs
     Subnet_mask=255.255.255.240
     MTU=1492
     Attaches_To_Area=0.0.0.3
     Hello_Interval=1
     Dead_Router_Interval=4
     Authentication_Key="xx";
OSPF_INTERFACE IP_Address=9.152.88.232
     Name=VIPAs
     Subnet_mask=255.255.255.248
     MTU=1492
     Attaches_To_Area=0.0.0.3
     Hello_Interval=1
     Dead_Router_Interval=4
     Authentication_Key="xx";
INTERFACE IP_Address=9.152.87.143
     Name=LINKETH
     Subnet_mask=255.255.248.0;
AS_Boundary_routing
     Import_Direct_Routes=YES
     Import_Static_Routes=YES;
```

Example 5-4 shows the result of a **`D TCPIP`** command.

Example 5-4 OSPF interfaces and neighbours of WPS3, that own the VIPAs

```
D TCPIP,,OMPR,OSPF,INTERFACE

EZZ7849I INTERFACES 591
IFC ADDRESS     PHYS         ASSOC. AREA     TYPE   STATE  #NBRS  #ADJS
9.152.88.35     EZASAMEMVS   0.0.0.3         P-2-MP    1      0      0
9.152.88.35     EZAXCFS8     0.0.0.3         P-2-MP    1      0      0
9.152.88.35     EZAXCFS6     0.0.0.3         P-2-MP    1      0      0
9.152.88.35     EZAXCFS2     0.0.0.3         P-2-MP    1      0      0
9.152.88.233    VIPL099858E9 0.0.0.3         VIPA    N/A    N/A    N/A
9.152.88.234    VIPL099858EA 0.0.0.3         VIPA    N/A    N/A    N/A
9.152.88.235    VIPL099858EB 0.0.0.3         VIPA    N/A    N/A    N/A
9.152.64.117    LINKVIA      0.0.0.3         BRDCST   32      3      2
9.152.88.35     EZAXCFS4     0.0.0.3         P-2-MP   16      1      1
D TCPIP,,OMPR,OSPF,NEIGHBOR
EZZ7851I NEIGHBOR SUMMARY 593
NEIGHBOR ADDR   NEIGHBOR ID     STATE  LSRXL DBSUM LSREQ HSUP IFC
9.152.64.116    9.152.94.197      128     0     0     0  OFF LINKVIA
9.152.64.118    9.152.64.118        8     0     0     0  OFF LINKVIA
9.152.64.115    9.152.94.193      128     0     0     0  OFF LINKVIA
9.152.88.36     9.152.64.118      128     0     0     0  OFF EZAXCFS4
```

Example 5-5 shows the OSPF configuration for the other system in the sysplex, in the test environment, *WPS4*.

Example 5-5 OSPF definition of system WPS4

```
RouterID=9.152.64.118;
```

```
Area
Area_Number=0.0.0.3
     stub_Area=YES
     Authentication_Type=Password;
OSPF_Interface IP_Address=9.152.64.118
     Name=LINKVIA
     Subnet_mask=255.255.255.240
     MTU=1492
     Attaches_To_Area=0.0.0.3
     Hello_Interval=1
     Dead_Router_Interval=4
     Authentication_Key="xx";
OSPF_Interface IP_Address=9.152.88.*
     Name=XCFs
     Subnet_mask=255.255.255.240
     MTU=1492
     Attaches_To_Area=0.0.0.3
     Hello_Interval=1
     Dead_Router_Interval=4
     Authentication_Key="xx";
OSPF_INTERFACE IP_Address=9.152.88.232
     Name=VIPAs
     Subnet_mask=255.255.255.248
     MTU=1492
     Attaches_To_Area=0.0.0.3
     Hello_Interval=1
     Dead_Router_Interval=4
     Authentication_Key="xx";
INTERFACE IP_Address=9.152.87.144
     Name=LINKETH
     Subnet_mask=255.255.248.0;
AS_Boundary_routing
     Import_Direct_Routes=YES
```

Example 5-6 shows the result of a **`D TCPIP`** command.

Example 5-6 OSPF interfaces and neighbours of WPS4

```
D TCPIP,,OMPR,OSPF,INTERFACE

EZZ7849I INTERFACES 969
IFC ADDRESS     PHYS         ASSOC. AREA     TYPE   STATE  #NBRS  #ADJS
9.152.88.36     EZASAMEMVS   0.0.0.3         P-2-MP    1      0      0
9.152.88.36     EZAXCFS8     0.0.0.3         P-2-MP    1      0      0
9.152.88.36     EZAXCFS6     0.0.0.3         P-2-MP    1      0      0
9.152.88.36     EZAXCFS3     0.0.0.3         P-2-MP   16      1      1
9.152.88.36     EZAXCFS2     0.0.0.3         P-2-MP    1      0      0
9.152.64.118    LINKVIA      0.0.0.3         BRDCST   32      3      2
D TCPIP,,OMPR,OSPF,NEIGHBOR
EZZ7851I NEIGHBOR SUMMARY 971
NEIGHBOR ADDR   NEIGHBOR ID     STATE  LSRXL DBSUM LSREQ HSUP IFC
9.152.88.35     9.152.64.117      128     0     0     0  OFF EZAXCFS3
9.152.64.117    9.152.64.117        8     0     0     0  OFF LINKVIA
```

```
9.152.64.116    9.152.94.197       128     0     0     0  OFF LINKVIA
9.152.64.115    9.152.94.193       128     0     0     0  OFF LINKVIA
```

5.2.5 DB2 consideration

DB2 Version 7 is configured in datasharing mode. On WPS3, the DB2 subsystem is started with the subsystem name SG91, and on WPS4, it is started with the subsystem name SG92. The database group name is DSN9 and the DB2 Location Name is WPSDSN9. For details on enabling datasharing mode, see *DB2 Data Sharing: Planning and Administration,* SC26-9935.

5.2.6 Shared HFS

The installation directory of WebSphere Portal (`WPS_PATH`) is made accessible by using the same directory name (`/usr/lpp/PortalServer`) on all systems in the sysplex.

You can see the shared HFS setup in Figure 5-2.

Note: The sysplex configuration of WebSphere Portal requires a **shared HFS** (`/PortalServer`) mounted in read/write mode and enabled by PARMLIB member BPXPRMxx SYSPLEX(YES).

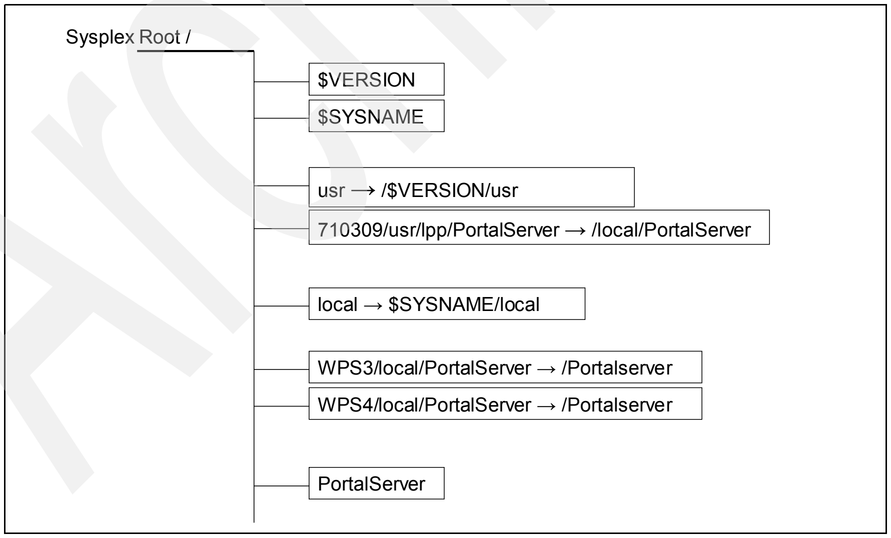
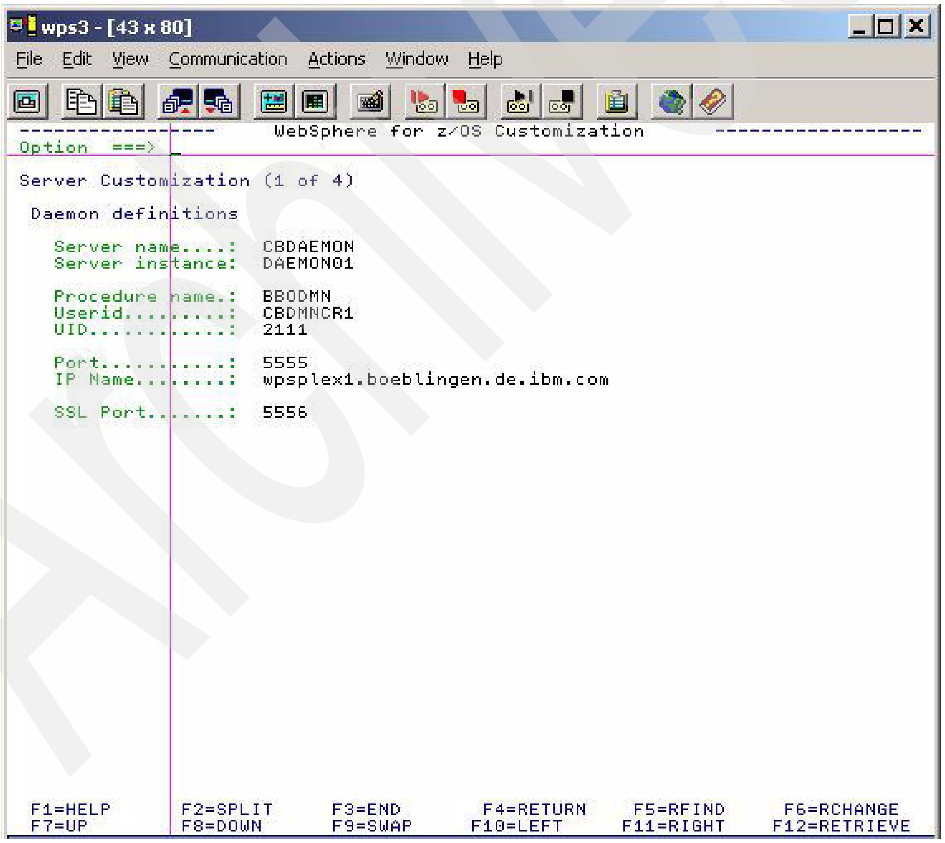

Sysplex Root /
$VERSION
$SYSNAME
usr → /$VERSION/usr
710309/usr/lpp/PortalServer → /local/PortalServer
local → $SYSNAME/local
WPS3/local/PortalServer → /Portalserver
WPS4/local/PortalServer → /Portalserver
PortalServer

Figure 5-2 Overview of the shared HFS of WebSphere Portal

5.2.7 Configuring WebSphere Application Server in a sysplex

WebSphere Application Server is bootstrapped to the first system (WPS3) in the sysplex and is extended to the other systems in the sysplex (WPS4). Use the SMEUI (Figure 5-4 on page 157) of WebSphere Application Server to configure it for each system in the sysplex. For more information, refer to *WebSphere Application Server Version 4.0.1 for z/OS and OS/390: Installation and Customization,* GA22-7834-06.

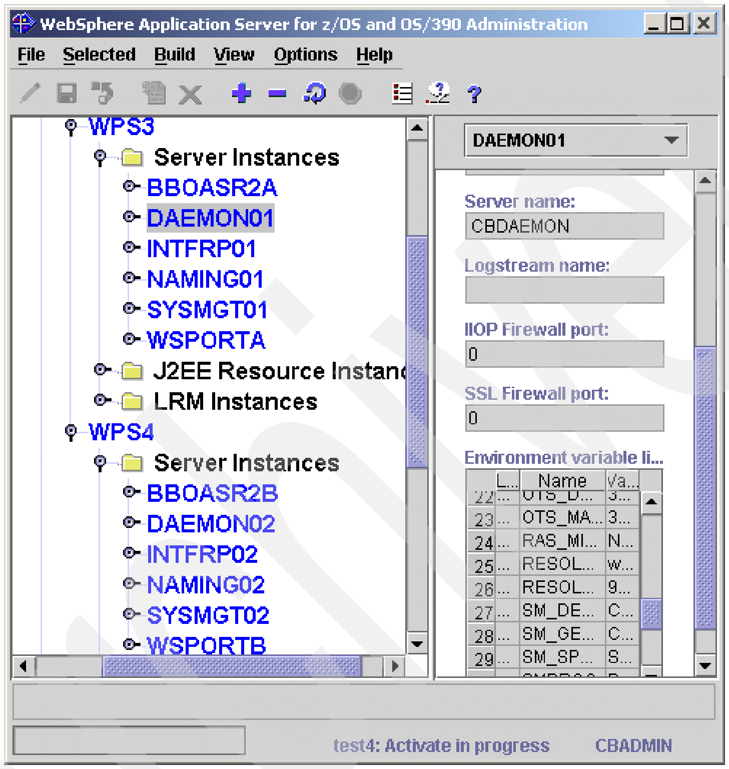
Note: Choose the IP daemon name carefully in the Customization dialog; it is very difficult to change after the bootstrap. In this sample, it is set on a dynamic VIPA. Tests have shown that if a static IP name is used, the daemon of the system that possesses this IP name must be started to use the SMEUI even to connect to another system in the sysplex. See Figure 5-3.



```
-----------------       WebSphere for z/OS Customization       ------------------
Option  ===> _

Server Customization (1 of 4)

 Daemon definitions

   Server name.....:   CBDAEMON
   Server instance:    DAEMON01

   Procedure name.:    BBODMN
   Userid.........:    CBDMNCR1
   UID............:    2111

   Port...........:    5555
   IP Name........:    wpsplex1.boeblingen.de.ibm.com

   SSL Port.......:    5556

 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 5-3 Customization dialog of WebSphere Application Server on WPS3

Note: We strongly recommend that you enable WebSphere Application Server in a sysplex before installing WebSphere Portal. This is because the installation scripts check how many systems are defined in WebSphere Application Server and create server instances for WebSphere Portal on each of them.


WebSphere Application Server for z/OS and OS/390 Administration

File Selected Build View Options Help

- WPS3
  - Server Instances
    - BBOASR2A
    - DAEMON01
    - INTFRP01
    - NAMING01
    - SYSMGT01
    - WSPORTA
  - J2EE Resource Instanc
  - LRM Instances
- WPS4
  - Server Instances
    - BBOASR2B
    - DAEMON02
    - INTFRP02
    - NAMING02
    - SYSMGT02
    - WSPORTB

DAEMON01

Server name: CBDAEMON

Logstream name:

IIOP Firewall port: 0

SSL Firewall port: 0

Environment variable li...

L...	Name	Va...
22...	OTS_D...	3...
23...	OTS_MA...	3...
24...	RAS_MI...	N...
25...	RESOL...	w...
26...	RESOL...	9...
27...	SM_DE...	C...
28...	SM_GE...	C...
29...	SM_SP...	S...

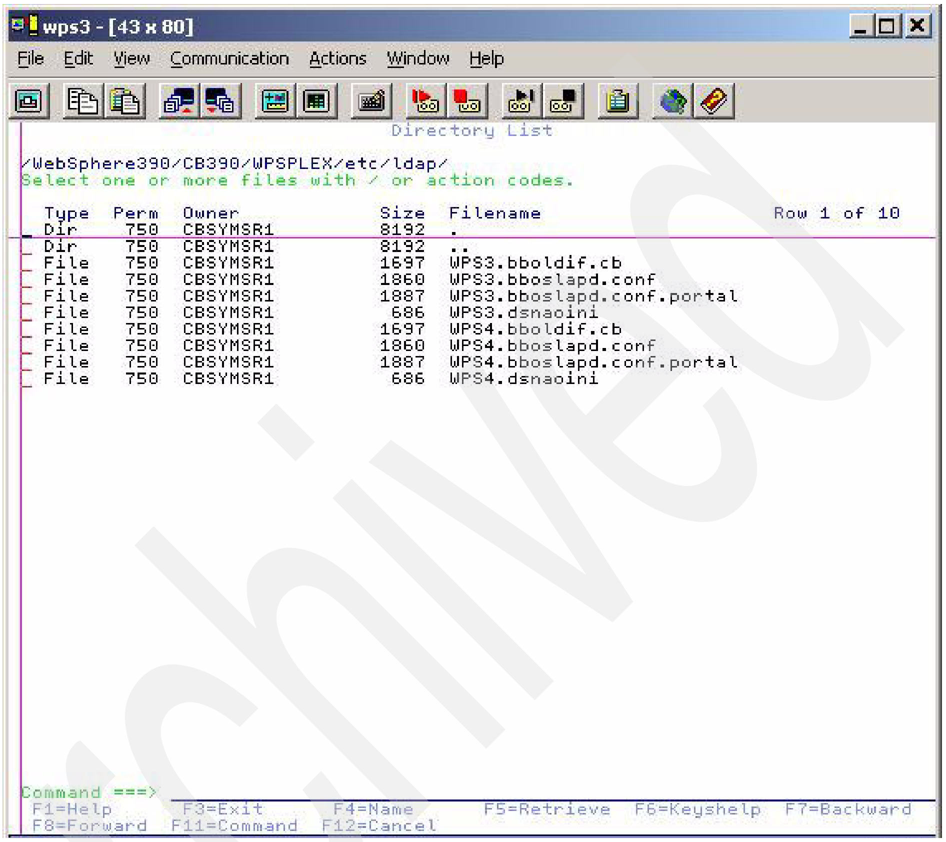
test4: Activate in progress CBADMIN

Figure 5-4 SMEUI for enabling WebSphere Application Server in the sysplex

5.2.8 Enable LDAP in the sysplex

Configure the LDAP settings in WebSphere Application Server for the sysplex. The customization of the LDAP settings of WebSphere Portal is the same as customizing the LDAP of WebSphere Application Server. For details, refer to *WebSphere Application Server Version 4.0.1 for z/OS and OS/390: Installation and Customization,* GA22-7834-06.

Figure 5-5 shows the LDAP configuration files that were copied from system WPS3 to WPS4 to enable the WebSphere Application Server LDAP and the Portal Server LDAP in a Sysplex.



```
wps3 - [43 x 80]
                         Directory List

/WebSphere390/CB390/WPSPLEX/etc/ldap/
Select one or more files with / or action codes.

  Type  Perm  Owner             Size  Filename                          Row 1 of 10
_ Dir    750  CBSYMSR1          8192  .
_ Dir    750  CBSYMSR1          8192  ..
_ File   750  CBSYMSR1          1697  WPS3.bboldif.cb
_ File   750  CBSYMSR1          1860  WPS3.bboslapd.conf
_ File   750  CBSYMSR1          1887  WPS3.bboslapd.conf.portal
_ File   750  CBSYMSR1           686  WPS3.dsnaoini
_ File   750  CBSYMSR1          1697  WPS4.bboldif.cb
_ File   750  CBSYMSR1          1860  WPS4.bboslapd.conf
_ File   750  CBSYMSR1          1887  WPS4.bboslapd.conf.portal
_ File   750  CBSYMSR1           686  WPS4.dsnaoini

Command ===>
 F1=Help      F3=Exit      F4=Name       F5=Retrieve  F6=Keyshelp  F7=Backward
 F8=Forward  F11=Command  F12=Cancel
```

Figure 5-5 Configuration files for enabling LDAP in the sysplex

Configure WebSphere Portal in a Sysplex

Here we discuss Configuring WebSphere Portal in a Sysplex in WebSphere Application Server. Installing WebSphere Portal in a sysplex environment is basically the same as installing it in a monoplex (refer to the PTF 3 Readme in Section “Installing WebSphere Portal for z/OS and OS/390 Version 4.1 PTF 3”), plus some additional setup. We recommend that you use setup values similar to the following sample configuration. WebSphere Portal is configured in a shared HFS that must be available for all systems.

1. Ensure that you have set up the configuration file of WebSphere Portal for a sysplex:

   `WPS_PATH/zoinst/scripts/wps.setup.in`

This should be done during an installation step of WebSphere Portal.

`WPS_PORT` is the HTTP Port of the webcontainer.

`WPS_RESOURCE_LOCATION` is the common location name of the DB2 subsystems.

`WPS_HOST` is the VIPA IP name that distributes requests for `WPS_ PORT`.

`WPS_SHORT_HOST` is the VIPA short name.

`WPS_LDAP_SERVER` is the URL of the LDAP server of WebSphere Portal. The LDAP of WebSphere Portal is also accessible through the sysplex distributor.

`WPS_LDAP_PORT` is the port of the LDAP of WebSphere Portal.

Example 5-7 shows the `wps.setup.in` file for the test environment.

Example 5-7 The wps.setup.in file for the test environment

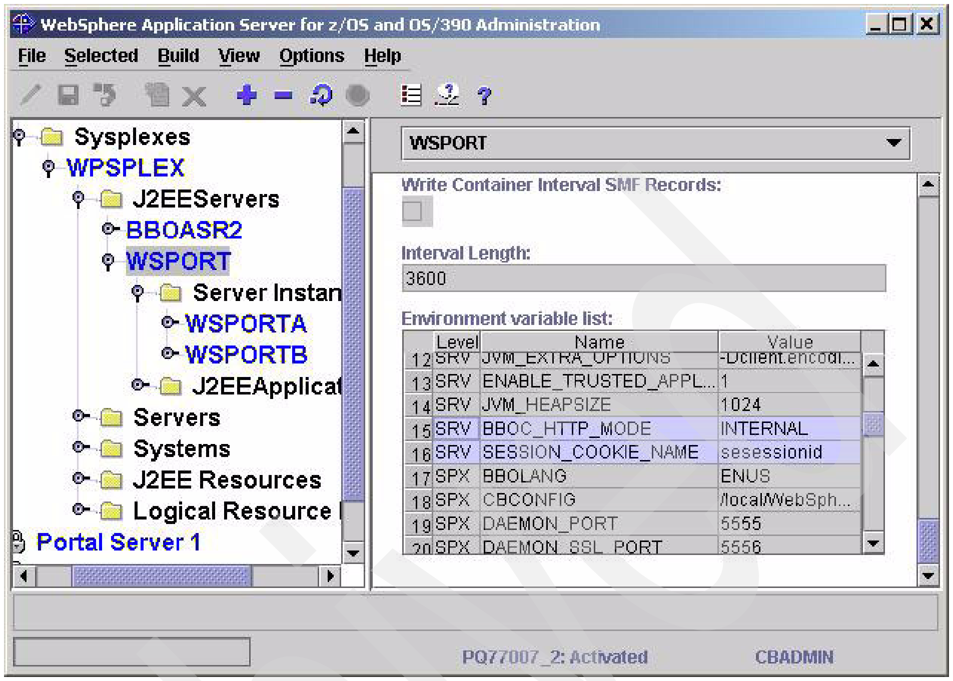
```
# HTTP port of the Portal Server. Required.
export WPS_PORT=8082
# Location name of the data base that should be used when establishing
# connections with the DB2 Datasource used by the Portal Server code.
# In a sysplex environment, MUST be identical on all systems. For
# details, refer to the "Installation Instructions" document
# Optional
export WPS_RESOURCE_LOCATION=WPSDSN9
# Name of the Portal Server host. Required for
# $WPS_PATH/libapp/config/services/ConfigService.properties
# and
# $WS_CONFIG_PATH/controlinfo/envfile/$SYSPLEX/$WPS_SERVERNAME/webcontainer.conf
export WPS_HOST=wpsplex2.boeblingen.de.ibm.com
# Short Name of the Portal Server host. Required for
# $WS_CONFIG_PATH/controlinfo/envfile/$SYSPLEX/$WPS_SERVERNAME/webcontainer.conf
export WPS_SHORT_HOST=wpsplex2
.
.
# Name of the LDAP server used by WebSphere Portal Server. Required for
# $WPS_PATH/libapp/config/um.properties
export WPS_LDAP_SERVER=wpsplex2.boeblingen.de.ibm.com

# Port of the LDAP server used by WebSphere Portal Server. Optional, if set,
# added to $WPS_PATH/libapp/config/um.properties
export WPS_LDAP_PORT=2389
```

Note: The variable `WPS_PORT` must be set to the HTTP port of the webcontainer during the WebSphere Portal installation. This is because the installation scripts set this as the `BBOC_HTTP_PORT` variable at server level during the configuration of WebSphere Application Server. Other network components, which can be used with WebSphere Application Server V4.0.1, such as Network Dispatcher, WebSphere Edge Server, and the WebSphere Advanced Edition HTTP plugin (introduced with WebSphere Application Server V4.0.1 PTFs UQ90051, UQ90052, and UQ70037) should be enabled subsequently.

2. Run the installation jobs delivered with WebSphere Portal. Refer to the PTF3 Readme (Section “Installing WebSphere Portal for z/OS and OS/390 Version 4.1 PTF3”), and Chapter 2, “Portal installation” on page 19. Make sure that WebSphere Portal is accessible with the URL you have specified in the file `wps.setup.in` during installation.
3. From the z/OS or OS/390 console, stop WebSphere Portal via the command: **`stop WSPORT.WSPORTA`**
4. After WebSphere Portal has been successfully installed, use the SMEUI of WebSphere Application Server to change the portal server instances for each additional system in the sysplex (WSPORTB, WSPORTC, and so on). You need to map the correct daemon and system management instances, and add environment variables at server instance level. For details, refer to *WebSphere Application Server Version 4.0.1 for z/OS and OS/390: Installation and Customization,* GA22-7834-06. Add two environment variables at server level (WSPORT):

```
BBOC_HTTP_MODE=INTERNAL
SESSION_COOKIE_NAME=sesessionid    (This is case sensitive.)
```

Use the SMEUI to add the variables as seen on Figure 5-6 on page 161.



Write Container Interval SMF Records:

Interval Length: 3600

Environment variable list:

	Level	Name	Value
12	SRV	JVM_EXTRA_OPTIONS	-Dclient.encodi...
13	SRV	ENABLE_TRUSTED_APPL...	1
14	SRV	JVM_HEAPSIZE	1024
15	SRV	BBOC_HTTP_MODE	INTERNAL
16	SRV	SESSION_COOKIE_NAME	sesessionid
17	SPX	BBOLANG	ENUS
18	SPX	CBCONFIG	/local/WebSph...
19	SPX	DAEMON_PORT	5555
20	SPX	DAEMON_SSL_PORT	5556

Figure 5-6 Defining environment variables using SMEUI

Note: Per WebSphere Application Server default, the session cookie name is JSESSIONID, in which case no environment variable SESSION_COOKIE_NAME needs to be specified. However, WebSphere Portal uses sesessionid as cookie name. Therefore, environment variable SESSION_COOKIE_NAME needs to be specified. The value matches the value session.cookie.name in file webcontainer.conf that is shipped and installed with WebSphere Portal.

5. Edit the configuration file to enable the HTTP plugin for WebSphere Portal:

   `/WPS_PATH/libapp/config/services/ConfigService.properties`

   The test environment specifies the URL and port of the IBM HTTP Server that contains the HTTP plugin and that is accessible through the sysplex distributor. The VIPA `wpsplex2.boeblingen.de.ibm.com` distributes requests for port 80 to which the IBM HTTP Server is listening.

```
# The parameters of the (virtual) host that the portal is accessed through
#
# Default: localhost (host.name = wpsplex2.boeblingen.de.ibm.com
host.name = wpsplex2.boeblingen.de.ibm.com
host.port.http = 80
host.port.https =
```

6. For each server instance of WebSphere Portal, edit the file webcontainer.conf in the directory:

   ```
   WS_CONFIG_PATH/controlinfo/envfile/SYSPLEXNAME/WPS_SERVERNAME*
   ```

   Where, * is A, B, C, and so on, depending on the number of systems in your sysplex. Add the host and port name of the IBM HTTP Server. If you have several HTTP servers with different names and ports where you are planning to run WebSphere Application Server plugins, add each of them for each server instance to the variable host.default_host.alias.

   The following is an example of the setup for server instance WSPORTA on system *WPS3*:

   ```
   host.default_host.alias=wpsplex2.boeblingen.de.ibm.com:8082,
   wpsplex2:8082,boewps3.boeblingen.de.ibm.com:8082, boewps3:8082,
   wpsplex2.boeblingen.de.ibm.com, wpsplex2
   ```

   The following is an example of the setup for server instance WSPORTB on system *WPS4*:

   ```
   host.default_host.alias=wpsplex2.boeblingen.de.ibm.com:8082,
   wpsplex2:8082,boewps4.boeblingen.de.ibm.com:8082, boewps4:8082,
   wpsplex2.boeblingen.de.ibm.com, wpsplex2
   ```

   Note that apart from the HTTP server host and port name **wpsplex2**, the home IP names of the systems **boewps3** and **boewps4 are also added.** This is done for test purposes only.

7. Enable the LDAP of WebSphere Portal in the sysplex, as described in 5.2.8, “Enable LDAP in the sysplex” on page 157, for each system on which you want to have this LDAP started. Enabling the portal LDAP is the same as enabling the LDAP of WebSphere Application Server in a sysplex.

   Start the portal LDAP server, for example, in the test environment:

   ```
   RO WPS3,S BBOLDAT
   ```

   ```
   RO WPS4,S BBOLDAT
   ```

Configure IBM HTTP Server and the HTTP Server Plugin

WebSphere Portal for z/OS and OS/390 supports the use of an external HTTP server or IBM HTTP Server on the same system where WebSphere Portal runs.

If you intend to implement such an HTTP setup with Network Dispatcher, WebSphere Edge Server, or WebSphere Advanced Edition HTTP plugin (introduced with WebSphere Application Server V4.0.1 PTFs UQ90051, UQ90052, and UQ70037), refer to redbook *Enabling High Availability e-Business on eServer zSeries,* SG24-6850.

The plugin used for this configuration is the new HTTP plugin for IBM HTTP Server running on z/OS and OS/390 that was introduced with PTF UQ74160 Service Level W401500. The plugin is required for routing HTTP requests.

The configuration files for the Web server, such as `httpd.conf` and `plugin-cfg.xml`, are kept separately on each system. They are staged in HFS path `/local/websrv`. As shown in Figure 5-2 on page 155, the path `/local` is resolved to `/$SYSNAME/local`. That means, there are actually two separate directories `/WPS3/local/websrv and /WPS4/local/websrv.`

1. Configure the HTTP server for the systems in the sysplex. Example 5-8 is a sample configuration for each IBM HTTP Server on WPS3 and WPS4 in the configuration file `httpd.conf`.

Example 5-8 httpd.conf file for WPS3 and WPS4

```
UserId PUBLIC
   .
   .
   .
# ======================================
# *** WAS W401500 PLUG-IN directives ***
# ======================================
ServiceSync On
#
ServerInit
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:init_exit
/local/websrv/plugin-cfg.xml
Service /wps/*
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:service_exit
Service /webapp/*
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:service_exit
Service /PolicyIVP/*
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:service_exit
ServerTerm
/usr/lpp/WebSphere/WebServerPlugIn/bin/ihs390WASPlugin_http.so:term_exit
   .
   .
   .
SSLClientAuth PASSTHRU
```

2. Configure the plugin-cfg.xml file. Example 5-9 shows the plugin configuration setup in the file plugin-cfg.xml for WPS3.

Example 5-9 Plugin-cfg.xml for WPS3

```
<?xml version="1.0"?>
<Config>
 <Log Name="/tmp/http/plugin.log" LogLevel="Trace" />
```

```
  <ServerGroup Name="IVPGroup">
   <ClusterAddress Name="dvipa2">
   <Transport Hostname="wpsplex2.boeblingen.de.ibm.com" Port="8081"
Protocol="http" />
   </ClusterAddress>
   <Server CloneID="BBOASR2.BBOASR2A" Name="BBOASR2A">
   <Transport Hostname="boewps3.boeblingen.de.ibm.com" Port="8081"
Protocol="http" />
   </Server>
   <Server CloneID="BBOASR2.BBOASR2B" Name="BBOASR2B">
   <Transport Hostname="boewps4.boeblingen.de.ibm.com" Port="8081"
Protocol="http" />
   </Server>
  </ServerGroup>
  <ServerGroup Name="PortalGroup">
   <ClusterAddress Name="dvipa2">
   <Transport Hostname="wpsplex2.boeblingen.de.ibm.com" Port="8082"
Protocol="http" />
   </ClusterAddress>
   <Server CloneID="WSPORT.WSPORTA" Name="WSPORTA">
   <Transport Hostname="boewps3.boeblingen.de.ibm.com" Port="8082"
Protocol="http" />
   </Server>
   <Server CloneID="WSPORT.WSPORTB" Name="WSPORTB">
   <Transport Hostname="boewps4.boeblingen.de.ibm.com" Port="8082"
Protocol="http" />
   </Server>
  </ServerGroup>
  <VirtualHostGroup Name="default_host">
   <VirtualHost Name="*:80"/>
   <VirtualHost Name="localhost:80"/>
   <VirtualHost Name="wpsplex2.boeblingen.de.ibm.com:80"/>
   <VirtualHost Name="boewps3.boeblingen.de.ibm.com:80"/>
  </VirtualHostGroup>
  <UriGroup Name="IVP_URIs">
   <Uri Name="/webapp/*"/>
   <Uri Name="/PolicyIVP/*"/>
  </UriGroup>
  <UriGroup Name="Portal_URIs">
   <Uri Name="/wps/*" affinityCookie="sesessionid"/>
  </UriGroup>
  <Route ServerGroup="IVPGroup" UriGroup="IVP_URIs"
   VirtualHostGroup="default_host"/>
  <Route ServerGroup="PortalGroup" UriGroup="Portal_URIs"
   VirtualHostGroup="default_host"/>
</Config>
```

Example 5-10 shows the plugin configuration setup in the file plugin-cfg.xml for WPS4.

Example 5-10   Plugin-cfg.xml for WPS4

```
<?xml version="1.0"?>
<Config>
 <Log Name="/tmp/http/plugin.log" LogLevel="Trace" />
  <ServerGroup Name="IVPGroup">
   <ClusterAddress Name="dvipa2">
   <Transport Hostname="wpsplex2.boeblingen.de.ibm.com" Port="8081"
Protocol="http" />
   </ClusterAddress>
   <Server CloneID="BBOASR2.BBOASR2A" Name="BBOASR2A">
   <Transport Hostname="boewps3.boeblingen.de.ibm.com" Port="8081"
Protocol="http" />
   </Server>
   <Server CloneID="BBOASR2.BBOASR2B" Name="BBOASR2B">
   <Transport Hostname="boewps4.boeblingen.de.ibm.com" Port="8081"
Protocol="http" />
   </Server>
  </ServerGroup>
  <ServerGroup Name="PortalGroup">
   <ClusterAddress Name="dvipa2">
   <Transport Hostname="wpsplex2.boeblingen.de.ibm.com" Port="8082"
Protocol="http" />
   </ClusterAddress>
   <Server CloneID="WSPORT.WSPORTA" Name="WSPORTA">
   <Transport Hostname="boewps3.boeblingen.de.ibm.com" Port="8082"
Protocol="http" />
   </Server>
   <Server CloneID="WSPORT.WSPORTB" Name="WSPORTB">
   <Transport Hostname="boewps4.boeblingen.de.ibm.com" Port="8082"
Protocol="http" />
   </Server>
  </ServerGroup>
   <VirtualHostGroup Name="default_host">
    <VirtualHost Name="*:80"/>
    <VirtualHost Name="localhost:80"/>
    <VirtualHost Name="wpsplex2.boeblingen.de.ibm.com:80"/>
    <VirtualHost Name="boewps4.boeblingen.de.ibm.com:80"/>
   </VirtualHostGroup>
   <UriGroup Name="IVP_URIs">
    <Uri Name="/webapp/*"/>
    <Uri Name="/PolicyIVP/*"/>
   </UriGroup>
   <UriGroup Name="Portal_URIs">
    <Uri Name="/wps/*" affinityCookie="sesessionid"/>
   </UriGroup>
   <Route ServerGroup="IVPGroup" UriGroup="IVP_URIs"
   VirtualHostGroup="default_host"/>
   <Route ServerGroup="PortalGroup" UriGroup="Portal_URIs"
   VirtualHostGroup="default_host"/>
</Config>
```

Note: WebSphere Portal is shipped and installed with:

```
session.cookie.name=sesessionid in webcontainer.conf
```

Therefore, specify affinityCookie=sesessionid in the plugin-cfg.xml file instead of the default JSESSIONID that WebSphere Application Server uses.

From the z/OS or OS/390 console, start WebSphere Portal on your systems via the command:

```
RO <yourSy1>,S WPS_PROCNAME.$WPS_SERVERNAMEA,srvname='$WPS_SERVERNAMEA'
RO <yourSy2>,S WPS_PROCNAME.$WPS_SERVERNAMEB,srvname='$WPS_SERVERNAMEB'
```

For example, in the test environment:

```
RO WPS3,S WSPORT.WSPORTA,srvname='WSPORTA'
RO WPS4,S WSPORT.WSPORTB,srvname='WSPORTB'
```

Start IBM HTTP Server, for example, in the test environment:

```
RO WPS3,S WEBSRV3
RO WPS4,S WEBSRV3
```

Connectivity

Figure 5-7 on page 167 illustrates how WebSphere Portal, IBM HTTP Server, and the sysplex distributor are connected and communicate in the test environment.

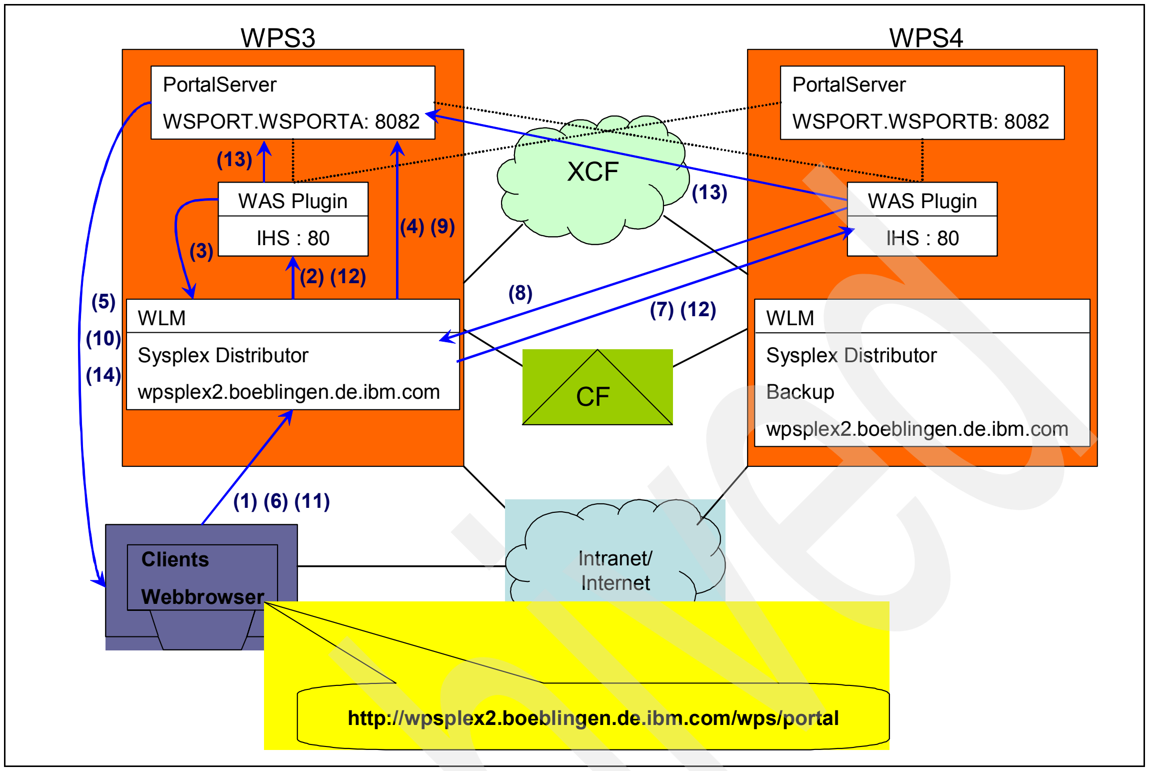


Figure 5-7 Connectivity in the test environment

The client request on `wpsplex2.boeblingen.de.ibm.com/wps/portal` is distributed by the sysplex distributor **(1)** to IBM HTTP Server on WPS3 **(2)**. The WebSphere Application Server plugin of IBM HTTP Server checks this HTTP request for session information. As there is no match with any entry that is configured in the file plugin-cfg.xml, it routes the request back to the sysplex distributor with port 8082 **(3)** because of a standard rule in plugin-cfg.xml that applies in that case.

```
<ClusterAddress Name="dvipa2">
<Transport Hostname="wpsplex2.boeblingen.de.ibm.com" Port="8082"
Protocol="http" />
```

In a case where a standard rule matches, the plugin trace shows:

```
st: htrequestGetCookie: Looking for cookie: 'sesessionid'
st: htrequestGetCookie: No cookie found for: 'sesessionid'
websphereParseSessionID: Parsing session id from '/wps/portal'
websphereParseSessionID: Failed to parse session id
websphereFindServer: Have a cluster address server so using it: dvipa2
websphereFindTransport: Finding the transport
```

```
websphereFindTransport: Setting the transport: wpsplex2.boeblingen.de.ibm.com
on port 8082
```

The sysplex distributor then routes the request to the server WSPORTA that is listening on port 8082 **(4)**. WSPORTA returns the login page to the client **(5)**. The client fills in the user ID and password, and sends the login page back to the sysplex distributor **(6)**. The sysplex distributor this time distributes the request to IBM HTTP Server on WPS4 **(7)**. The WebSphere Application Server plugin routes the request back to the sysplex distributor because the standard rule matches again **(8)**. The sysplex distributor distributes the request to server WSPORTA on WPS3 **(9)** and WSPORTA returns a cookie and a welcome message **(10)**. The user is now logged on, and any subsequent request to WebSphere Portal contains session information in the request header. Like before, subsequent requests first get distributed on port 80 by the sysplex distributor **(11)** to an IBM HTTP Server **(12)**. Because there is session information in the request header, the WebSphere Application Server plugin of IBM HTTP Server directly routes the request to WSPORTA **(13)**, which answers directly **(14)**.

The following rule matches if the HTTP request contains session information.

```
<Server CloneID="WSPORT.WSPORTA" Name="WSPORTA">
<Transport Hostname="boewps3.boeblingen.de.ibm.com" Port="8082" Protocol="http"
</Server>
```

When the rules match, the HTTP plugin trace shows:

```
websphereParseSessionID: Parsing session id from 'msp=2;
w3ibmProfile=2000120605...
sesessionid=0000ar4zaJBx-NTcMwqh8nCFUm0:WSPORT.WSPORTA'
websphereParseCloneID: Adding clone id 'WSPORT.WSPORTA'
websphereParseCloneID: Returning list of clone ids
group: serverGroupFindClone: Looking for clone
group: serverGroupFindClone: Match for clone 'WSPORTA'
websphereHandleSessionAffinity: Setting server to WSPORTA
websphereFindTransport: Finding the transport
websphereFindTransport: Setting the transport: boewps3.boeblingen.de.ibm.com on
port 8082
```

5.3 Parallel sysplex scenarios and tests

We ran some tests using our sysplex setup. We show the configuration we used and then we define scenarios for testing.

5.3.1 Sysplex configuration used for the tests

When you have completed the sysplex configuration as described in 5.2, “Setting up the infrastructure” on page 148, the system is similar to that illustrated below, which is a high level view of this configuration.

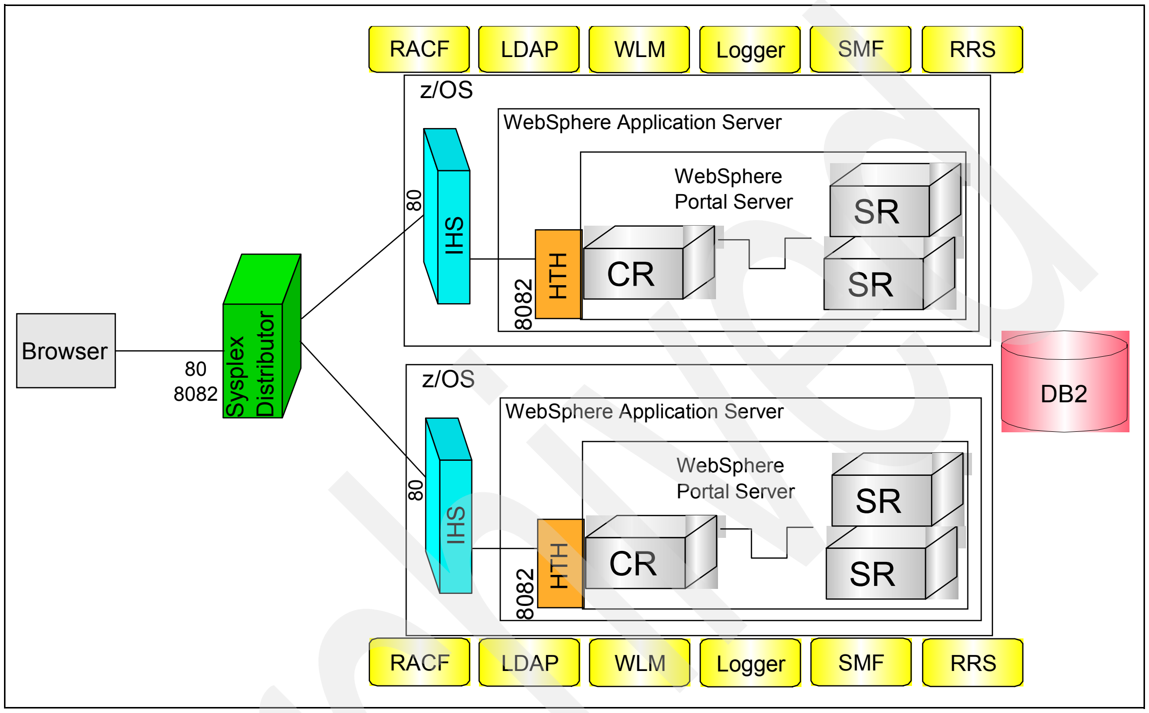


Figure 5-8 Sample sysplex WPSPLEX

Two z/OS systems (WPS3 and WPS4) are configured in a sysplex (WPSPLEX). Three sysplex distributors (WPSPLEX1, WPSPLEX2, and WPSPLEX3) are configured in the sysplex. The DB2 tables and file systems are shared with read/write access in the sysplex.

Figure 5-9 on page 170 shows details of WebSphere Application Server, particularly the servers that are started in WebSphere Application Server on each system. It also shows details of the shared database and file systems.

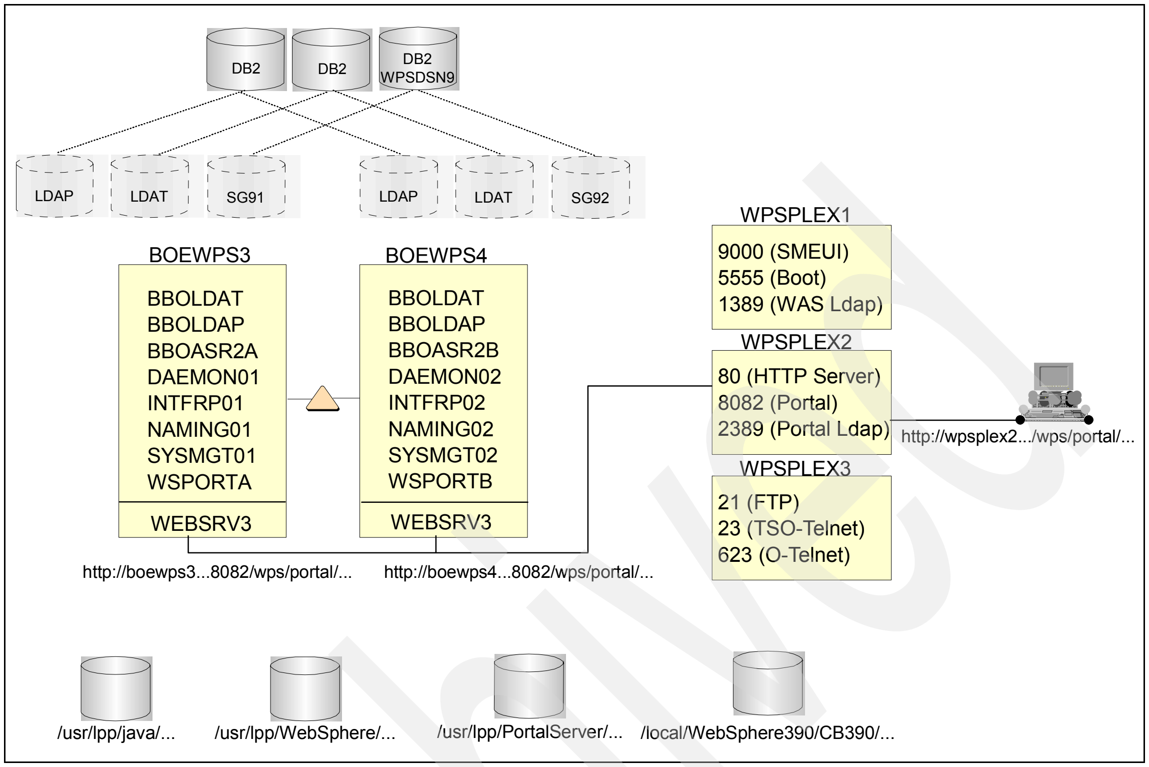


Figure 5-9 WebSphere Application Server in a sysplex configuration

The behavior of WebSphere Portal in a sysplex is primarily defined by the support that WebSphere Application Server provides for its applications. These supporting functions can be combined in a few ways within the sysplex to enable parallel processing on a single system.

5.3.2 Review of the system components in the sysplex

The following terms are used when explaining various types of parallel processing. This is similar to their usage in the documentation of WebSphere Application Server.

- **z/OS system**: A computer and its associated devices where z/OS or OS/390 and WebSphere Application Server for z/OS are running.
- **Sysplex**: A set of z/OS or OS/390 systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. A sysplex is a single-image system complex. It is one or more z/OS or OS/390 systems that, together, provide a single system complex (for example, it may be two LPARs on the

same hardware). This means that while a sysplex might be composed of multiple systems, it acts and reacts like a single instance.

- **Server**: A logical grouping of server instances. All server instances within a server are identical in structure and run the same set of applications.

  Administration is usually done at the server level. In addition, a server, from a management perspective, is a single entity in the sysplex. The server presents a single interface to the network and operator for control.

- **Server instance**: A functional component on which WebSphere Application Server for z/OS applications runs. It is an instance of a replicated server that can provide all the functions that the server makes available. All server instances within a server are identical in structure.

  A server instance has two kinds of address spaces: a Control Region (CR) and one or more Server Regions (SR). Application server code runs in a server region. A server region can be replicated based on the workload demands of the system. The control regions queue messages to the server regions.

5.3.3 Parallel processing scenarios

WebSphere Application Server offers settings to enable parallel processing both on a single system and in a sysplex. The most important settings that were used to define parallel scenarios are explained here.

- **(Server) Isolation policy**: One transaction or multiple transactions per server region?

  You have a choice of running server regions with an isolation policy of one transaction per server region or multiple transactions per server region. From a performance perspective, running more threads in a server region will consume less memory but at the cost of thread contention. This contention is application dependent. It is generally recommended that you use multiple transactions unless you have contention problems.

- **(Server) Replication policy**: One or multiple server regions per server?

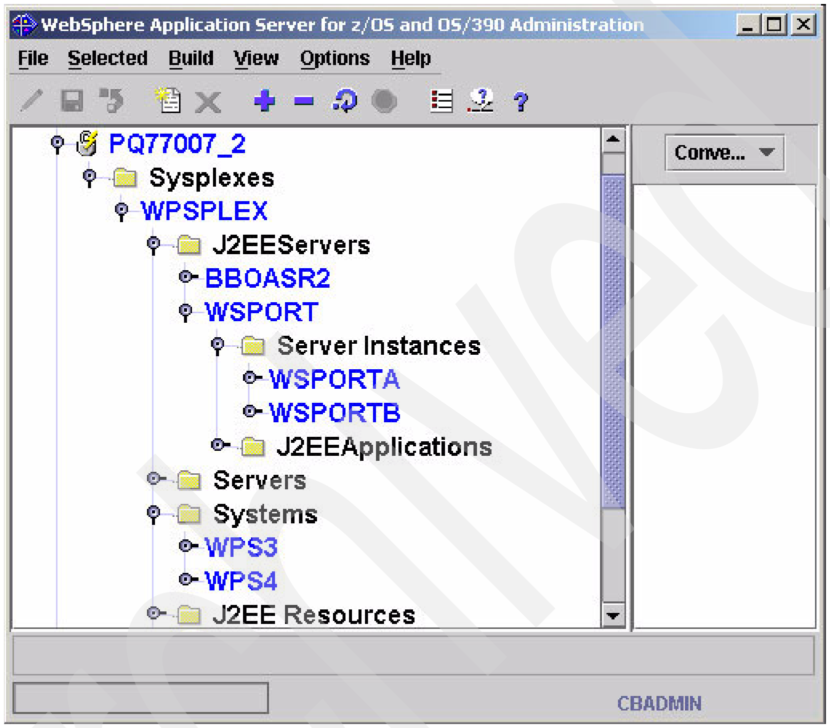
  The replication policy specifies how many server regions should be started. This is used when the server regions are replicated inside the server. The default value is to replicate as needed.

  One server region per server is recommended for transient objects. If you select multiple server regions per server, you can define a minimum and maximum boundary for the number of server regions to be started. Specify this using the variables `MAX_SRS=<#> and MIN_SRS=<#>` in the SMEUI of WebSphere Application Server at server level.

- Multiple server instances on distinct systems in a sysplex

  To create this setup, perform the configuration as described in 5.2, “Setting up the infrastructure” on page 148.

Figure 5-10 shows the resulting configuration in the SMEUI.


WebSphere Application Server for z/OS and OS/390 Administration
File
Selected
Build
View
Options
Help
Conve...
CBADMIN

```
PQ77007_2
  Sysplexes
    WPSPLEX
      J2EEServers
        BBOASR2
        WSPORT
          Server Instances
            WSPORTA
            WSPORTB
          J2EEApplications
      Servers
      Systems
        WPS3
        WPS4
      J2EE Resources
```

Figure 5-10 Sysplex in SMEUI

For more information, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration,* SA22-7835.

As a combination of the isolation policy, replication policy, and multiple server instances, the following scenarios were created. The summary of the scenarios can be seen in Table 5-1 on page 173.

Scenario 1: Reference scenario with no parallel processing of incoming requests

One WebSphere Portal instance only was started (one control region in one system, and one server region defined per control region). A single transaction per server region was defined.

Scenario 2: Parallel processing of incoming requests within one server region

One WebSphere Portal instance only was started (one control region in one system, and one server region defined per control region). Multiple transactions per server region were defined.

Scenario 3: Horizontal scaling

This is similar to the scenario referred to as *horizontal scaling* in the documentation of WebSphere Portal for Multiplatforms. One WebSphere Portal instance only was started (one control region in one system, and two server regions defined per control region). Multiple transactions per server region were defined.

Scenario 4: Two server instances on one system

This scenario is untested, because there was no additional benefit when compared to scenario 3, assuming that an exact clone is required.

Scenario 5: Vertical scaling

This scenario is for parallel processing in a sysplex. It is similar to the scenario referred to as *vertical scaling* in the documentation of WebSphere Portal for Multiplatforms. Two WebSphere Portal instances were started (one control region in each system, and one server region defined per control region). Multiple transactions per server region were defined.

Table 5-1 Parallel Scenarios - Summary Table:

Scenario Description	**Scenario**				
	1	**2**	**3**	**4**	**5**
Single transaction per server region	X				
Multiple transactions per server region (recommended by WebSphere Application Server)		X	X	X	X
1 server region per server (recommended by WebSphere Application Server for transient objects)	X	X		X	X
Multiple server regions per server (like horizontal scaling in WebSphere Portal for Multiplatforms)			X		

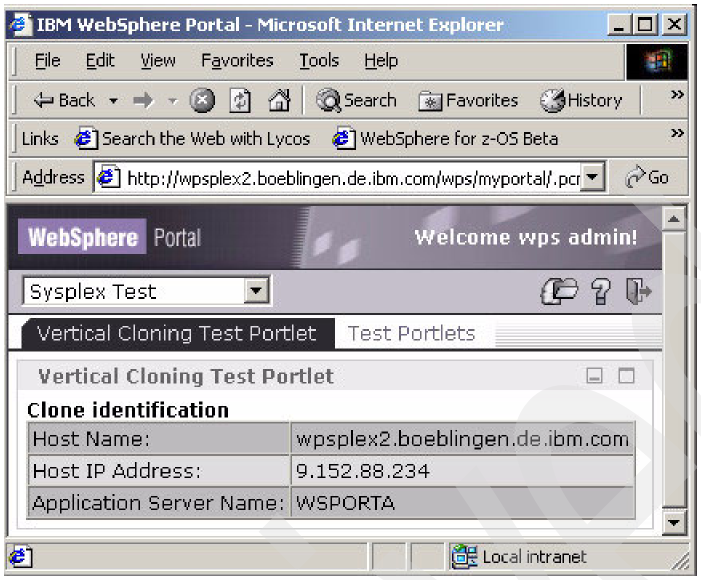
1 server instance	X	X	X		
2 server instances 1 system (not tested because reasonable usage was not determined)				X	
Multiple server instances in multiple systems (sysplex) (like vertical scaling in WebSphere Portal for Multiplatforms)					X
All scenarios were executed using session affinity and no session persistence.					

5.3.4 The test cases

The main intention for the test is to verify that WebSphere Portal on z/OS in a sysplex behaves like WebSphere Portal for Multiplatforms when the latter uses horizontal or vertical scaling. The intent was not to exploit all possible scenarios within the sysplex on z/OS, for example, session persistence and failover scenarios were not tested.

One problem during these tests was to show in which server instance of the portal a special scenario was executed. Two tools were used to solve this problem: the Vertical Cloning Test Portlet and a modified Login class that writes trace messages.

The Vertical Cloning Test Portlet that is used in the class IBM WebSphere Portal Version 4 Administration (WebSphere Portal for Multiplatforms) was installed. This portlet was deployed without change on the z/OS portal and displayed the Clone identification, as shown in Figure 5-11 on page 175.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Search the Web with Lycos WebSphere for z-OS Beta
Address http://wpsplex2.boeblingen.de.ibm.com/wps/myportal/.pcr
Go
WebSphere Portal
Welcome wps admin!
Sysplex Test
Vertical Cloning Test Portlet
Test Portlets
Local intranet

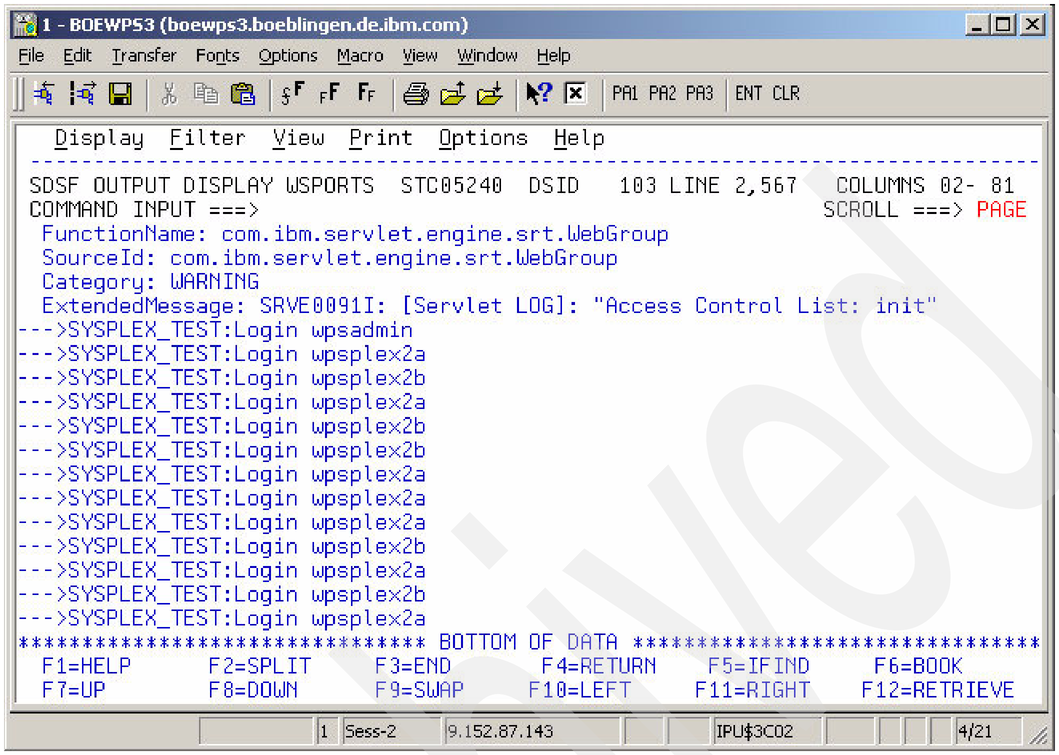
Vertical Cloning Test Portlet

Clone identification

Host Name:	wpsplex2.boeblingen.de.ibm.com
Host IP Address:	9.152.88.234
Application Server Name:	WSPORTA

Figure 5-11 Vertical Cloning Test Portlet

The Vertical Cloning Test Portlet was helpful during browser tests. It was not used during the automated test scenarios or during parallel scenarios 1 to 3, because the Host Name, Host IP Address, and Application Server Name are identical for all clones in these scenarios. To cover these cases, the Login class was modified to print a trace statement (`--->SYSPLEX_TEST:Login <userid>`) in the standard output each time a login was made. This modification produced similar traces to those shown in Figure 5-12 on page 176.


1 - BOEWPS3 (boewps3.boeblingen.de.ibm.com)
File Edit Transfer Fonts Options Macro View Window Help
PA1 PA2 PA3 ENT CLR
Display Filter View Print Options Help

```
 SDSF OUTPUT DISPLAY WSPORTS  STC05240  DSID   103 LINE 2,567   COLUMNS 02- 81
 COMMAND INPUT ===>                                             SCROLL ===> PAGE
  FunctionName: com.ibm.servlet.engine.srt.WebGroup
  SourceId: com.ibm.servlet.engine.srt.WebGroup
  Category: WARNING
  ExtendedMessage: SRVE0091I: [Servlet LOG]: "Access Control List: init"
--->SYSPLEX_TEST:Login wpsadmin
--->SYSPLEX_TEST:Login wpsplex2a
--->SYSPLEX_TEST:Login wpsplex2b
--->SYSPLEX_TEST:Login wpsplex2a
--->SYSPLEX_TEST:Login wpsplex2b
--->SYSPLEX_TEST:Login wpsplex2b
--->SYSPLEX_TEST:Login wpsplex2a
--->SYSPLEX_TEST:Login wpsplex2a
--->SYSPLEX_TEST:Login wpsplex2a
--->SYSPLEX_TEST:Login wpsplex2b
--->SYSPLEX_TEST:Login wpsplex2a
--->SYSPLEX_TEST:Login wpsplex2b
--->SYSPLEX_TEST:Login wpsplex2a
******************************** BOTTOM OF DATA ********************************
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 5-12 Standard output showing the login trace

Manual test with a browser

Log in manually via a browser to each of the portal instances. In all cases, the login can be done using the same or different user IDs. The user IDs must have the necessary access rights to access the security portlet. The access rights of some user IDs are changed on one system. The users logged in to the other portal instances verified that the modified rights are visible in all other portal instances.

Automated scenarios

An automated scenario is created with AK/Stress to log in, view some portlets, and log out again.

One user ID only used by all clients

Logging in from four clients with one user ID resulted in a successful scenario, but produced this exception in all scenarios (except scenario 1), as expected:

```
com.ibm.wps.util.ConcurrentModificationException
```

The exception is produced because if you use only one user ID with four automated clients, the probability arises that two or more clients try to access the same database fields at about the same time.

Distinct user IDs used by each client

Logging in from four clients parallel with different user IDs resulted in successful scenarios, without the exception:

```
com.ibm.wps.util.ConcurrentModificationException.
```

Portlet deployment test

WebSphere Portal on z/OS and OS/390 uses a modified procedure for portlet deployment that is in part different to that of WebSphere Portal for Multiplatforms (for details, refer to "Portlet deployment" on page 178). During the portlet deployment test, two portlets were deployed. The first part of the deployment of each portlet was done in one of the portal instances. The second part of the deployment was done on one system in the sysplex. The result was that after the second part was completed, both portlets were available in each portal instance.

5.3.5 Considerations for parallel processing of multiple portal instances

When you run multiple Portal instances you should be aware of some shared resources and their uses. Here we give some advice on shared resources.

Shared resources

For all the parallel scenarios, all resources (file systems, property files, logs, and DB2, and so on) are shared between the portal server instances that run in parallel, regardless of whether the instances run on one or multiple systems.

Portal database access

If you log in to the portal two times with the same user ID or when different users concurrently use portlets that access the same backend data, the same database records are used and modified. The portal database is set up to allow maximum concurrency with data integrity (isolation level Cursor Stability). Database rows are locked for access during a very short period only. This means that one process may read database fields and another process is able to modify the fields immediately after the read is complete.

For example, if you use the same user ID during multiple logins (regardless of whether you use multiple instances of the portal) you might see error messages like the one shown below. This message might also appear if you have only one instance of the portal and you log in more than once.

```
2003.07.30 10:54:41.441 com.ibm.wps.datastore.UserDescriptorPersister
handleException
```

```
  Error during database access! Last SQL statement: {UPDATE USER_DESC SET
MODIFIED = ?, LAST_LOGIN = ? WHERE ( (OID = ?) AND (MODIFIED = ?) )}
2003.07.30 10:54:41.441 com.ibm.wps.datastore.core.DataStoreContext
handleException
  com.ibm.wps.util.ConcurrentModificationException: Database has been changed
since creation of data object! ObjectID = 10
   at
com.ibm.wps.datastore.core.BasePersister.storeUpdate(BasePersister.java:208)
```

Log files and standard output files

For the tests above:

- All property files are shared.
- The standard output file is written to the job log of each server region (each instance of the portal).
- The portal log file (wps<timestamp>.log in directory WPS_PATH/log) is written separately for each server region (each instance of the portal).
- The logs of WebSphere Transcoding Publisher (TranscoderMessagesx.log and TranscoderTracex.log) are written once only. All entries from all portal instances are accumulated into these files.

Portlet deployment

Portlet deployment is performed in two parts, as described in the topic “Deploying portlets on z/OS and OS/390” in the InfoCenter of WebSphere Portal for z/OS and OS/390.

Administrative tasks on WebSphere Portal using the administration portlet for installing portlets. This step generates job files with descriptions of further administration steps to be done in WebSphere Application Server. Changes to the portal configuration are effective immediately but the portlets are operational only after the next step. These tasks can be executed by any instance of the portal. The generated job files are all written into the same directory in shared HSF.

Administrative tasks are performed on WebSphere Application Server. Here, you install new portlets or update or delete existing ones by deploying them using the `WPAConfig` script and the jobs created by the administration portlet. The `WPAConfig` script can be executed on any z/OS system of the sysplex. It searches for jobs created by the administration portlet and executes them one by one in the sequence of their generation. During this step, any job files from all portal instances will be processed. After the successful execution of the tasks above, the portlets are available on each instance of the portal.

Important: During the second phase of the deployment, WebSphere Application Server stops and restarts each WebSphere Portal instance (stops each control region and the associated server regions). During the tests above, the stop and restart cycle frequently failed. To avoid this situation, stop all WebSphere Portal instances before you start the `WPAConfig` script (or the corresponding JCL procedure).

Summary

The three tests above were successfully executed for the parallel processing scenarios 1, 2, 3, and 5. The outcome was as expected, that is, the same as for WebSphere Portal for Multiplatforms, including the production of this exception in test case 2:

```
com.ibm.wps.util.ConcurrentModificationException
```

In addition, we have seen that if WebSphere Application Server is configured in the sysplex, it is very easy to install WebSphere Portal. There are no differences between portlet deployment in a single system and portlet deployment in a sysplex.

All migration steps from PTF2 to PTF3 that were executed were done on one system in the sysplex. All WebSphere Portal instances were successfully migrated, without any change in the migration procedure compared to migration in a single system installation.

6

Portlet development and deployment

This chapter deals with portlet development and deployment.

First we discuss the use of WebSphere Studio Application Developer (Application Developer) for the development of portlets for z/OS. Note: The installation of Application Developer is described in Chapter 3, “Development environment installation” on page 93.

Next we describe the steps involved in deploying portlets on the z/OS platform, highlighting the major differences between portlet deployment on the z/OS platforms versus the distributed platforms.

Then, we provide examples of deploying portlets using the following portlets:

- Document viewer
- Click-to-Action
- Host on Demand
- Lotus iNotes™
- Changing a Web application to a portlet to access a CICS® application

6.1 Portlet development

Although portlets can be developed using a simple text editor and the Java Development Toolkit, the tool of choice for portlet developers is WebSphere Studio Application Developer (Application Developer); see reference:

http://www-3.ibm.com/software/awdtools/studioappdev/

In addition the Bowstreet Portlet Factory available from the Portlet Catalog NavCode® 1WP10009Y can be used with Application Developer to provide rapid portlet development using a library of portlet components. Section 6.4.3, “Installing portlets obtained from the Portlet Catalog” on page 215 provides more information about the Portlet Catalog.

Application Developer V4.0.3 for Microsoft Windows is provided as part of the development environment for WebSphere Portal on z/OS. The development environment using Application Developer can be configured to include WebSphere Application Server AEs and WebSphere Portal Server for the debugging of portlets within Application Developer.

The IBM Portal Toolkit provides the capabilities to customize and manage the enterprise portal and create, test, debug and deploy individual portlets and Web content. Templates enable developers to quickly and easily create their own portlets. The Portal Toolkit plugs into Application Developer to provide a comprehensive framework for the development of e-business portal applications.

This installation and setup of Application Developer with WebSphere Application Server AEs, WebSphere Portal Server and Portal Toolkit is described in Chapter 3, “Development environment installation” on page 93.

Portlets overview

Portlets are reusable components that provide access to Web-based content, applications, and other resources. Companies can create their own portlets or select portlets from a catalog of third-party portlets. Portlets are intended to be assembled into a larger portal page, with multiple instances of the same portlet displaying different data for each user.

From a user's perspective, a portlet is a window on a portal site that provides a specific service or information, for example, a calendar or news feed. From an application development perspective, portlets are plug-able modules that are designed to run inside a portlet container of a portal server.

The portlet container provides a runtime environment in which portlets are instantiated, used, and finally destroyed. Portlets rely on the portal infrastructure to access user profile information, participate in window and action events, communicate with other portlets, access remote content, lookup credentials, and to store persistent data. The Portlet API provides standard interfaces for these functions, and is implemented by an extension of the servlet API.

IBM is working with other companies to standardize the Portlet API, making portlets interoperable between portal servers that implement the specification. The Portlet API offered in WebSphere Portal Version 4.1 is the first step toward the Portlet API standardization. For more information about the portlet specification, see:

`http://jcp.org/jsr/detail/168.jsp`

The Portlet API is described in detail in the InfoCenter at:

`http://publib.boulder.ibm.com/pvc/wp/current/ena/en/InfoCenter/index.html`

Portlets run inside the portlet container on the portal server, which is able to use services provided by the servlet container of the application server. Therefore, the portal container relies on the J2EE architecture implemented by WebSphere Application Server. As a result, portlets are packaged in WAR files similar to J2EE Web applications and are deployed like servlets. Like other servlets, a portlet is defined to the application server using the servlet deployment descriptor (`web.xml`). This file defines the Portlet’s class file and read-only initialization parameters.

Portlet development

The *Portlet Development Guide* is available from:

`http://www7b.software.ibm.com/wsdd/zones/portal/portlet/portletdevelopmentguide.html`

It provides an overview, examples of developing portlets, as well as using JSPs for presentation, and is recommended reading for portlet developers.

Note: The Portlet Development Guide has been updated for WebSphere Portal V4.2 on Multiplatforms. Although some of the examples are not applicable to WebSphere Portal Enable V4.1 for z/OS it is still recommended reading for portlet developers.

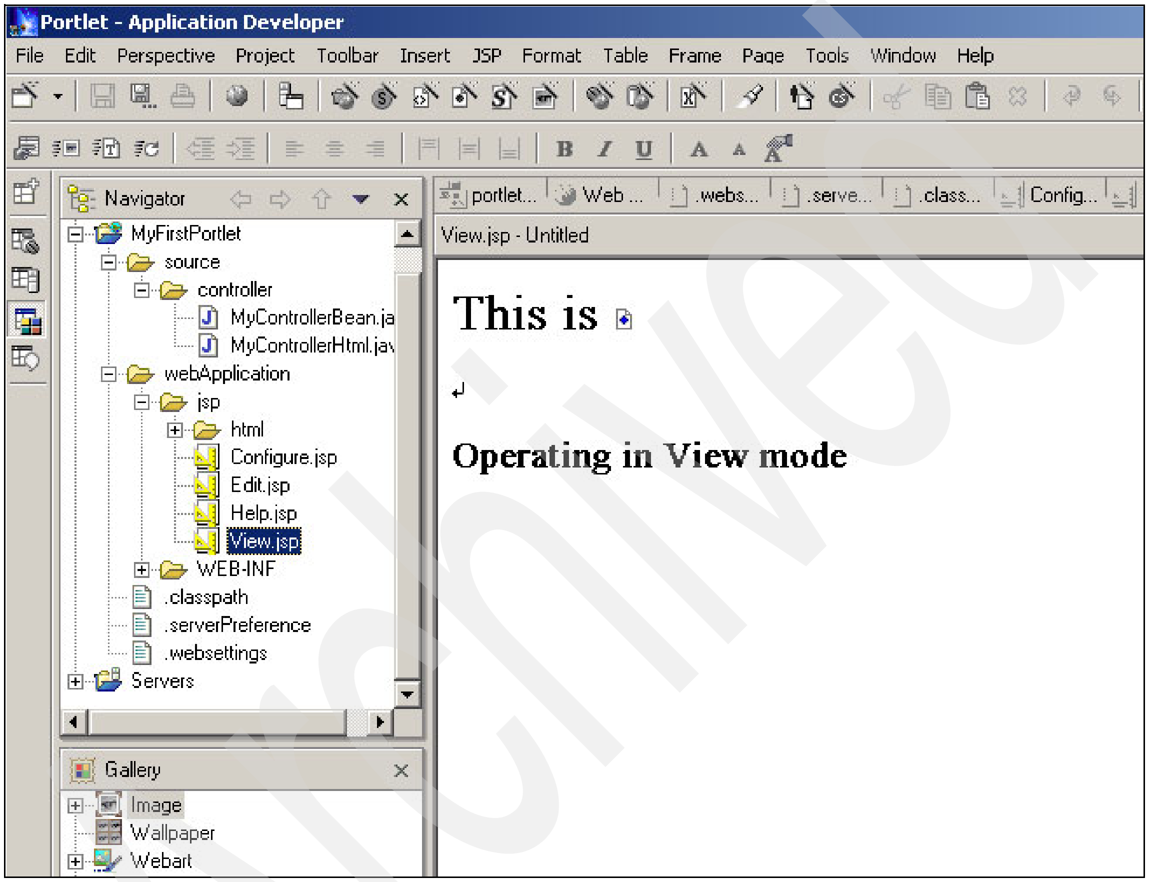
Portlets with WebSphere Portal Enable on z/OS

Many of the portlets available for WebSphere Portal Enable V4.1 for z/OS are described in the document *`Viewing the WebSphere Portal V4.1 Portlets`* available at:

`http://www7b.software.ibm.com/wsdd/library/techarticles/0301_son/0301_son.html`

6.1.1 Using Application Developer to develop a portlet

In “Installing the Portal Toolkit” on page 104 we used WebSphere Studio Application Developer to develop MyFirstPortlet to test the development and debug environment of Application Developer as shown in Figure 6-1.



This is

Operating in View mode

Figure 6-1 MyFirstPortlet in WebSphere Studio, View.JSP

This portlet did not require the developer to add or provide any custom Java code, and used two Java beans to execute the MVC model of the portlet, and several JSPs to display the text output.

The IBM WebSphere Developer Domain Portal Zone and WebSphere Studio Zones are found at:

http://www7b.software.ibm.com/wsdd/

These zones provide many useful articles on developing portlets including the obligatory *Hello Worl*d example. The portlet Java code for this is shown in Example 6-1:

Example 6-1 Simplest Hello world portlet example

```
package com.ibm.portlets.sample;

//portlet APIs
import org.apache.jetspeed.portlet.*;
import org.apache.jetspeed.portlets.*;

//Java stuff
import java.io.*;

public class HelloWorld extends AbstractPortlet {

  public void service(PortletRequest request, PortletResponse response)
    throws PortletException, IOException {

    PrintWriter writer = response.getWriter();
    writer.println("<p>Hello World!</p>");
  }
}
```

WebSphere Developer Domain article *Developing and Invoking a Servlet from a Portlet using WebSphere Studio* is found at:

http://www7b.software.ibm.com/wsdd/techjournal/0302_konduru/konduru.html

This article describes how to develop a portlet using Application Developer and use the portlet to invoke a servlet also developed using Application Developer.

6.2 Portlet development example

In this section we show how to use Application Developer to develop the *Hello World* portlet described in the previous section, and we export it for deployment to WebSphere Portal Server on z/OS.

Start Application Developer and from the menu bar, and select **Perspective** -> **Open-Other** as shown in Figure 6-2 on page 186.

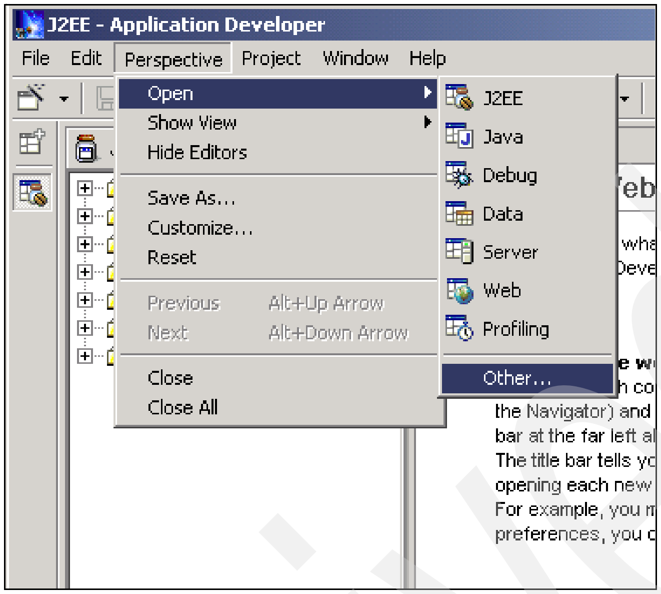


Figure 6-2 Opening Perspective Other

From the Select Perspective window select **Portlet** as shown in Figure 6-3.

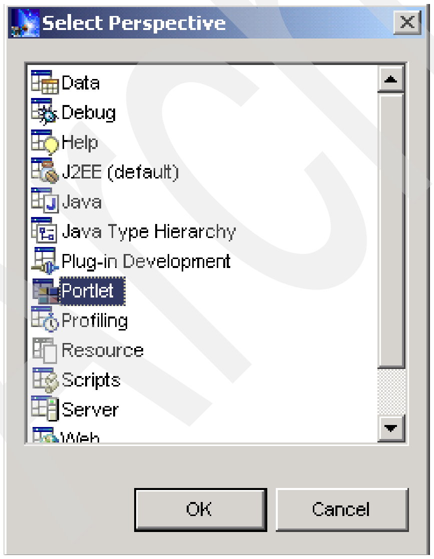


Figure 6-3 Select the Portlet Perspective

Right-click in the **Portlet Perspective Navigator** frame and select **New->Other**, as shown in Figure 6-3.

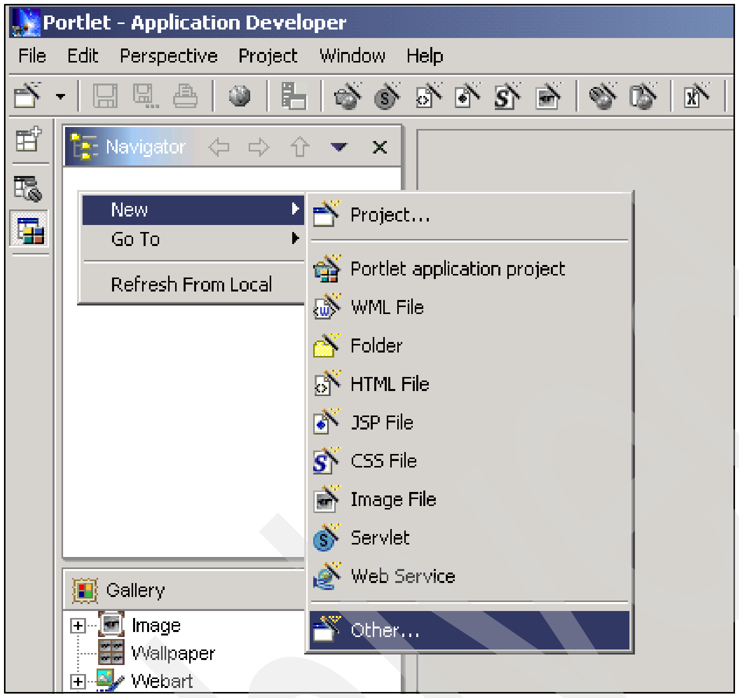


Figure 6-4 Select Other for type of Project

In the New project window select **`Portlet development`** in the left frame and **`Portlet application project`** in the right frame as shown in Figure 6-5 on page 188, and then click **`Next`**.

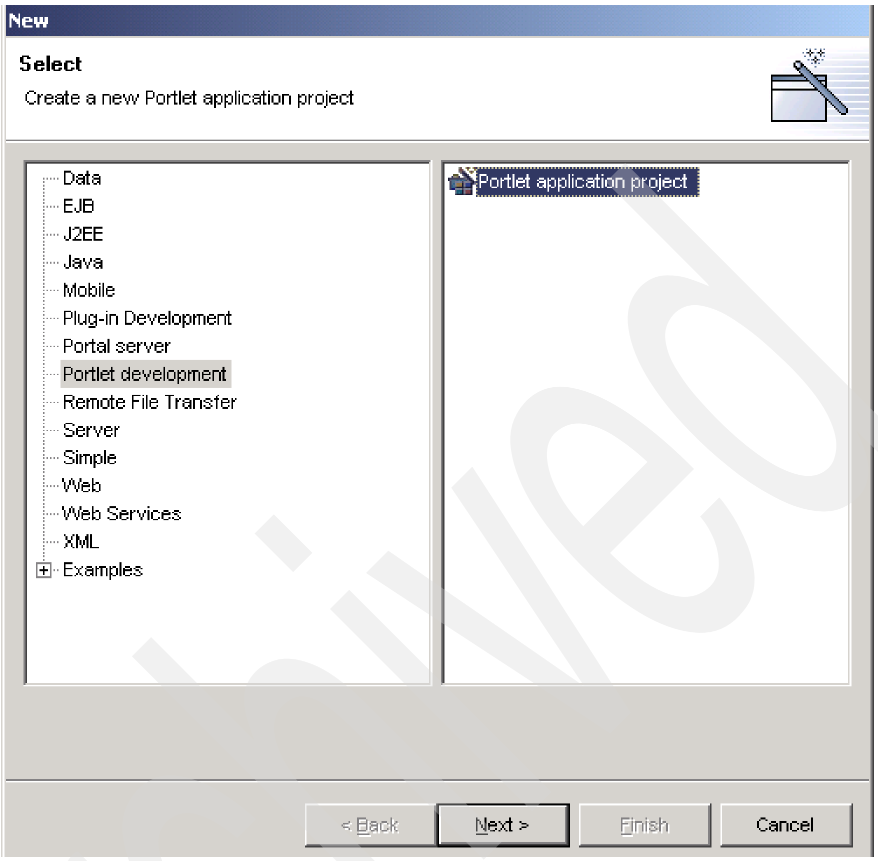
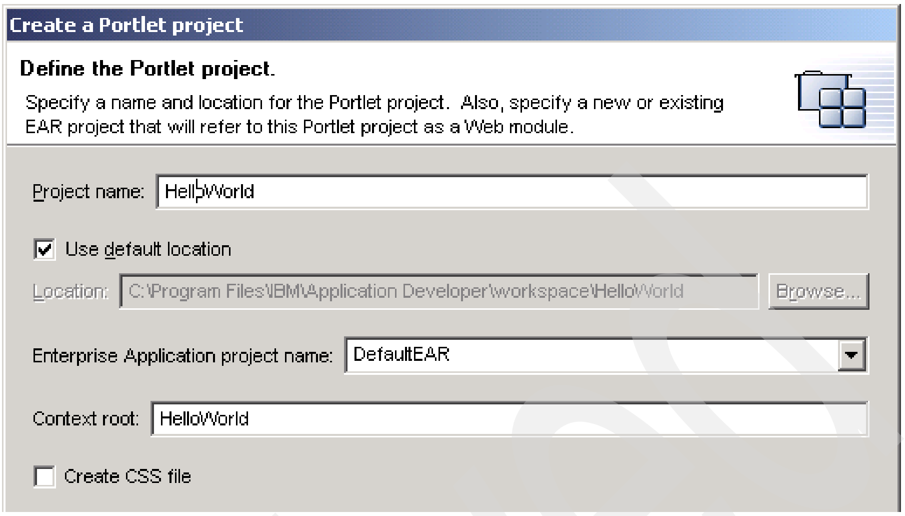


Figure 6-5 Select to create a new Portlet application project

On the next window you need to enter settings that define the project name and location, the EAR file to be used and the URI context root. The settings we used are shown in Figure 6-6 on page 189. We clicked on **Next** to continue at this stage, and not **Finish.**


Create a Portlet project

Define the Portlet project.

Specify a name and location for the Portlet project. Also, specify a new or existing EAR project that will refer to this Portlet project as a Web module.

Project name: HelloWorld

[✓] Use default location

Location: C:\Program Files\IBM\Application Developer\workspace\HelloWorld

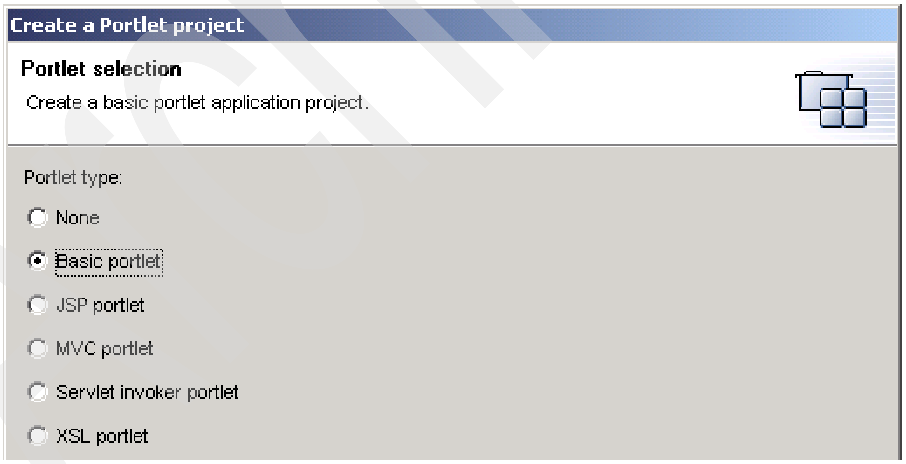
Enterprise Application project name: DefaultEAR

Context root: HelloWorld

[ ] Create CSS file

Figure 6-6 Define the project settings

The next window allows you to define the type of portlet to be used. Since this was going to be our first portlet with our own Java code using Application Developer, we chose a Basic portlet as shown in Figure 6-7. Click on **Next** to continue.


Create a Portlet project

Portlet selection

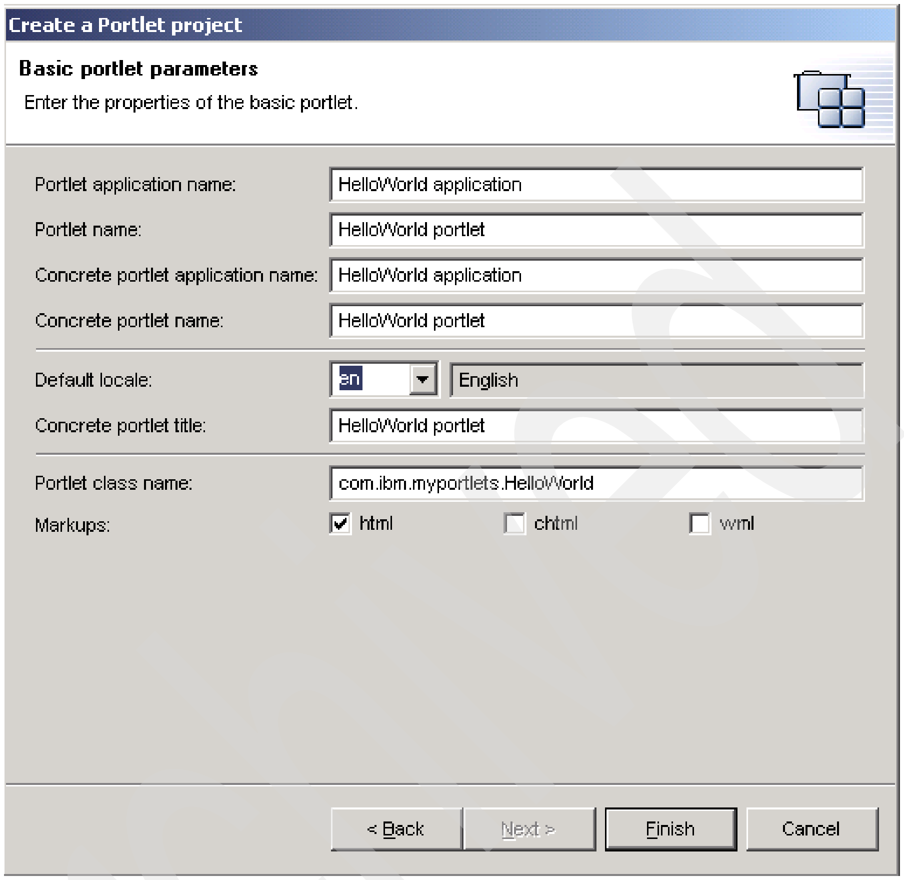
Create a basic portlet application project.

Portlet type:

None [✓ Basic portlet] JSP portlet MVC portlet Servlet invoker portlet XSL portlet

Figure 6-7 Choose the type of portlet to be created

The basic set of portlet parameters next need to be defined, and we chose the ones shown in Figure 6-8 on page 190, and then clicked on **Finish**.


Create a Portlet project

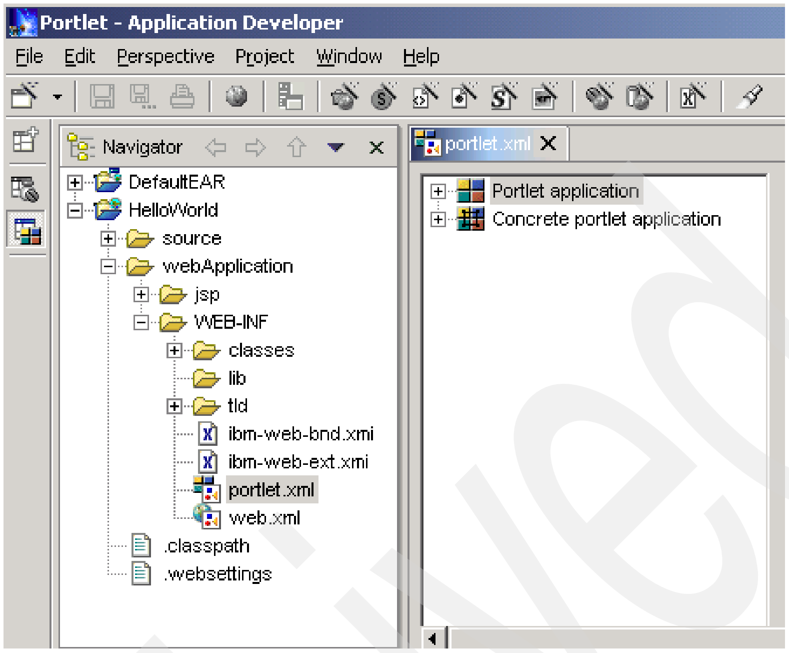
Basic portlet parameters

Enter the properties of the basic portlet.

Portlet application name:	HelloWorld application
Portlet name:	HelloWorld portlet
Concrete portlet application name:	HelloWorld application
Concrete portlet name:	HelloWorld portlet
Default locale:	en English
Concrete portlet title:	HelloWorld portlet
Portlet class name:	com.ibm.myportlets.HelloWorld
Markups:	[✓ html] chtml wml

Figure 6-8 Hello World portlet parameters

After clicking on **Finish**, Application Developer now builds the project which takes a few moments during which you will see several messages displayed as it creates directories, files, compiles Java code, until the project is created. The project is shown in the portlet perspective as seen in Figure 6-9 on page 191.


Portlet - Application Developer
File Edit Perspective Project Window Help
Navigator
portlet.xml

```
DefaultEAR
HelloWorld
    source
    webApplication
        jsp
        WEB-INF
            classes
            lib
            tld
            ibm-web-bnd.xmi
            ibm-web-ext.xmi
            portlet.xml
            web.xml
    .classpath
    .websettings
```

```
Portlet application
Concrete portlet application
```

Figure 6-9 The Hello World project

To add the Java code we expanded the source tree in the Navigator panel of the Hello World project by selecting the “+” next to **`HelloWorld`** -> **`source`** -> **`com`** -> **`ibm`** -> **`myportlets`**, as shown in Figure 6-10 on page 192.

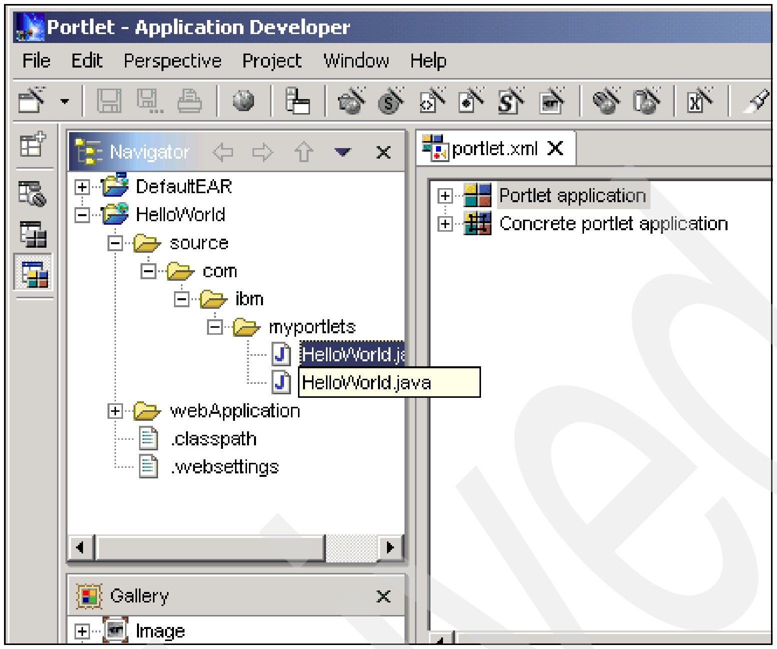
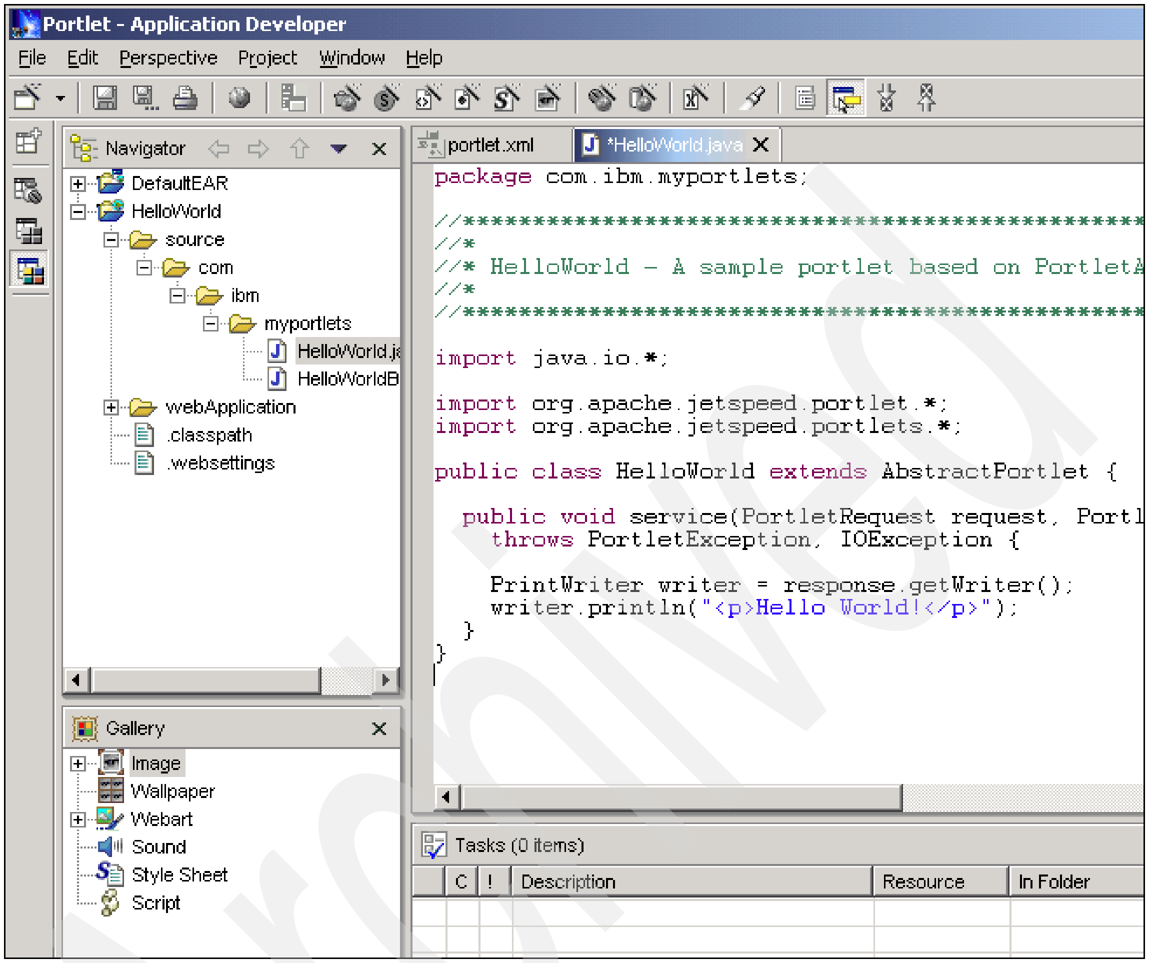


Figure 6-10 Expanded source tree

By double-clicking on **`HelloWorld.java`** we obtained the source view of the Java code that Application Developer had already generated for our project. Since this was our first Hello World java portlet we deleted all the generated code and instead copied the code from Example 6-1 on page 185.

We then changed the Java package to the portlet class name we had defined in “Hello World portlet parameters” on page 190 as shown in Figure 6-11 on page 193.

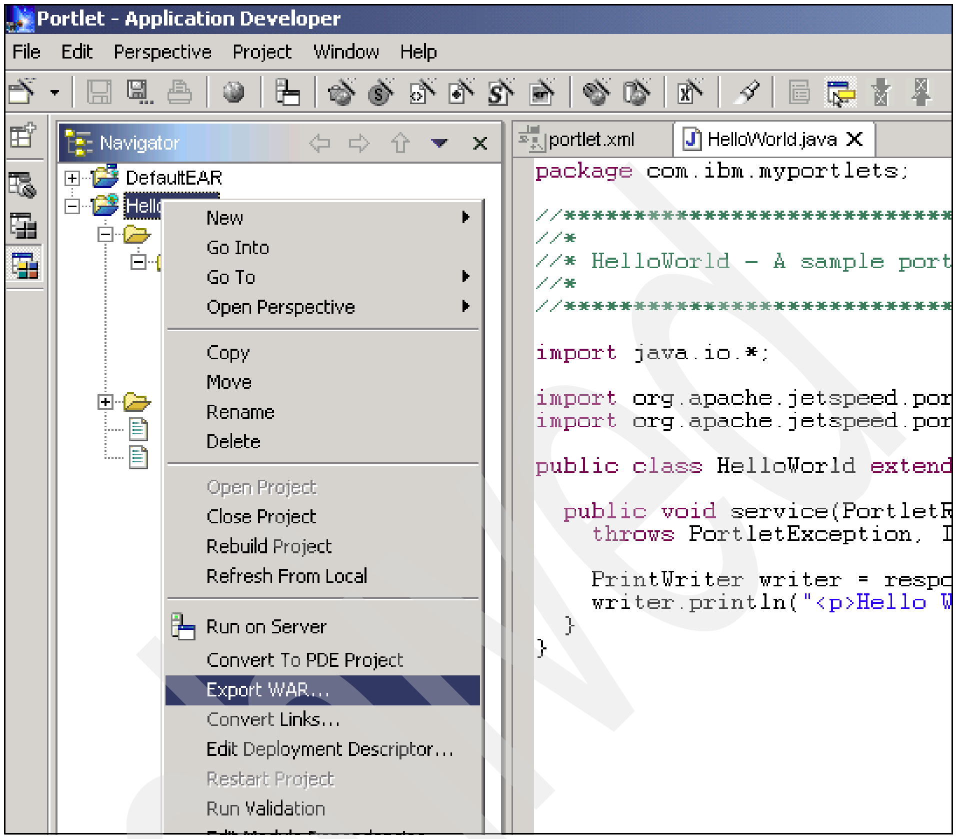


```
package com.ibm.myportlets;

//**********************************************
//*
//* HelloWorld - A sample portlet based on PortletA
//*
//**********************************************

import java.io.*;

import org.apache.jetspeed.portlet.*;
import org.apache.jetspeed.portlets.*;

public class HelloWorld extends AbstractPortlet {

  public void service(PortletRequest request, Portl
    throws PortletException, IOException {

    PrintWriter writer = response.getWriter();
    writer.println("<p>Hello World!</p>");
  }
}
```

Figure 6-11   Hello World java

Using the keyboard we entered **Ctrl+s**, which causes Application Developer to build the project and save it. If there are no errors listed in the tasks frame at the bottom right of the Application Developer window then the portlet project has been built successfully and can be tested and deployed.

By right-clicking on the project name in the Navigator frame, and selecting Export WAR, as shown in Figure 6-12 on page 194, the portlet WAR file can be exported from Application Developer to a local directory of choice for deployment on WebSphere Portal for z/OS.



```
package com.ibm.myportlets;

//******************************
//*
//* HelloWorld - A sample port
//*
//******************************

import java.io.*;

import org.apache.jetspeed.por
import org.apache.jetspeed.por

public class HelloWorld extend

  public void service(PortletR
    throws PortletException, I

    PrintWriter writer = respo
    writer.println("<p>Hello W
  }
}
```

Figure 6-12 Export WAR for the Hello World portlet

Deployment of portlets is described in the next section and we will deploy out Hello World portlet in 6.4.4, “Deploy the Hello World Portlet” on page 222.

6.3 Portlet deployment

After the portlet application is developed and its portlets packaged into a Web Application Archive (WAR) file, it is ready for deployment. In the context of WebSphere Portal Server, it means installing the portlet on the server for users to place onto a portal page.

Portlet deployment on the z/OS platform is a two step process that performs a configuration change to the Portal Server and additionally performs a change to the WebSphere Application Server.

To deploy portlets in WebSphere Portal on z/OS and OS/390 platform, you perform the following steps.

1. Administrative tasks on WebSphere Portal using the administration portlets for installing portlets. These require a user with portal administrative rights to login via a browser from the portal home page. This portlet deployment step generates job files in the z/OS Portal HFS directory structure for further administration steps to be done to WebSphere Application Server. Changes to the portal configuration are effective immediately but the portlets are not operational until the second step is performed.

   Note: This is different from portlet deployment on a distributed platform which does not have the second step.

2. Administrative tasks on WebSphere Application Server, where you install, update or delete portlets by running script **`WPAConfig`** and the jobs created in Step 1. The WPAConfig script searches for the jobs and executes them one by one in the sequence of their generation. After this step the portlets are operational on WebSphere Portal server.

Tip: If the Portal Server is running during Step 2 when executing script **`WPAConfig`**, it is stopped and restarted for each of the portlet deployment jobs it finds. Therefore, if you are deploying a large number of portlets from within a WAR file, we recommend you stop the Portal Server first, execute **`WPAConfig`** using one of the methods described later in this chapter, and then restart the Portal Server.

6.3.1 Portlet deployment jobs

When the administration portlets in WebSphere Portal server are used to install, delete, or update a portlet they generate portlet deployment job files on the HFS file system on z/OS. These files contain descriptions of further administrative steps to be done in WebSphere Application Server. There is one file for each configuration step, which is referred to as a *job*. Usually this file is accompanied by further resources, such as a file containing the portlet binaries. The job descriptions in the job files consist of name-value pairs that are recognized by the WPAConfig script. The generated job files only support application type or apptype of *war*.

Note: Although the *ear* apptype is recognized by WPAConfig script, it is officially not supported.

Table 6-1 shows the name-value pairs that WPAConfig recognizes and the context in which they apply.

Table 6-1 Name-value pairs that WPAConfig script recognizes

Name	Value	Required for action
action	CreateApp UpdateApp DeleteApp	All actions
apptype	war ear	CreateApp and UpdateApp
appsrc	File name relative to the job directory. It is usually the name of a WAR or EAR file	CreateApp and UpdateApp
context	Context Root value needed when generating an EAR file to wrap a WAR file	CreateApp and UpdateApp required only if apptype=war
appname	Display name of the application needed when generating an EAR file to wrap a WAR file. For the DeleteApp action, this is the name of the application to delete.	DeleteApp. CreateApp and UpdateApp required only if apptype=war
servername	Server Name to which the WebSphere Portal was deployed. For ex. WSPORT	All actions

6.3.2 Directory structure for portlet deployment jobs

Portlet deployment jobs are stored in a file hierarchy that is rooted in the portal temporary directory. The temporary directory is determined during installation of WebSphere Portal.

Note: During installation of our system the following variables were set up for the directory structure, and are referred to throughout the rest of this chapter:

- WPS_PATH: /usr/lpp/PortalServer/PortalServer
- WS_PATH: /usr/lpp/WebSphere
- WS_CONFIG_PATH: /WebSphere390/CB390

The script `wpsPre.root.sh`, found in <WPS_PATH>/PortalServer/zosinst/scripts, writes the name of the temporary directory into WPAConfig script. It also writes it into the ConfigService.properties file found in this directory, by setting the property wps.deployment.jobs:

```
<WPS_PATH>/PortalServer/libapp/config/services
```

The default location for this portal temporary directory is <WPS_PATH>/temp. The jobs directory is located in this temporary directory and contains five sub-directories. The name and contents of those sub-directories are shown in Table 6-2:

Table 6-2 Name and contents of deployment sub-directories

Directory name	Directory Contents
date	Generated by the administration portlet contains the jobs by date in the format yyyy_mm_dd. A job directory contains all the files that belong to a single job including the job file itself and the portlet's binary file that needs to be installed or updated.
done	Contains jobs and resources that have been successfully executed. Jobs are stored in the hierarchy when they were first generated.
failed	Contains all those jobs that were not successfully executed. Jobs are stored in the hierarchy when they were first generated.
work	Contains all those jobs that the script found in the *jobs* directory at the beginning of execution and moved to *work* directory.
log	For each run, the script writes a log file. These log files have the same name as the conversation that is used to make changes to WebSphere Application Server. This name consists of a prefix, the date, and the time of execution.

In our project environment, for example, WebSphere Portal was installed in /usr/lpp/PortalServer/PortalServer.

So the jobs file hierarchy was as follows:

```
/usr/lpp/PortalServer/PortalServer/temp/jobs
/usr/lpp/PortalServer/PortalServer/temp/jobs/done/<date1-directory>
/usr/lpp/PortalServer/PortalServer/temp/jobs/failed<date1-directory>
/usr/lpp/PortalServer/PortalServer/temp/jobs/log/<WPConversation_date1_time
1.log>
/usr/lpp/PortalServer/PortalServer/temp/jobs/work/<date1-directory>
```

6.3.3 The Portlet deployment script

As mentioned, WPAConfig.sh is the name of the script and is found in the <WPS_PATH>/PortalServer/zosinst/scripts directory.

While processing a portlet deployment job, the WPAConfig script:

1. Parses the arguments. If the optional parameter that determines the number of jobs to process is incorrect, it writes an error message and exits.
2. Builds global variables that are used throughout the script.
3. Searches for new jobs in the jobs directory and moves them into the *work* directory.
4. Looks in the *work* directory and checks for jobs. If there are no jobs, it exits.
5. Creates a new conversation.
6. Performs a loop over all the jobs or over as many jobs as specified by the optional parameter:
   a. Analyzes the job file to see what action is required.
   b. If the action is CreateApp or UpdateApp, resolves the WAR or EAR file and deploys the portlet into the application server.
   c. If the action is DeleteApp, removes the portlet from the application server
   d. Moves successful jobs to the *done* directory.
   e. If the deployment of an individual job fails, it moves the failed job to the *failed* directory.
7. Closes the conversation.
8. Exits the script.

Every step of the script is documented with messages that are written to the standard output and to a log file. When the script finishes, it exits with a return code that indicates whether or not it was successful. The return codes are shown in Table 6-3.

The WPAConfig script has one optional parameter, which must be a positive integer. This parameter determines the maximum number of jobs that are processed by the script. If the parameter is not specified, the script will process all available jobs.

Table 6-3 WPAConfig script return codes

Return code	Meaning
0	Everything was successfully run.
4	There were warnings.
8	There were errors.

6.3.4 Portlet deployment steps

There are two methods to execute WPAConfig.sh script when deploying portlets. The first is to run the WPACONF job that was used during installation to deploy the portlets installed with Portal Server, and the other way is to use the WPAConfig.sh script by telnet'ing to the z/OS UNIX environment.

Portlet deployment using a TSO session

The steps to deploy a portlet using the WPACONF job are:

1. Bring up the login portal page by entering the following URL in a Web browser: `http://<HOST_NAME:8082>/wps/portal`.
2. Log on to WebSphere Portal as a user with administrator rights, and via the Portal Administration GUI, install, delete, or update portlets as required. Make sure this step completes successfully.
3. Logon to a TSO session and stop WebSphere Portal via the command:

   `/p <WPS_PROCNAME>.<WPS_SERVERNAME>A.`

   For example, we used: **`/p WSPORT.WSPORTA`**.
4. Submit the WPACONF job to deploy the portlets to the hosting WebSphere Application Server to make them operational. The JCL for this job was customized during installation; refer to Chapter 2, "Portal installation" on page 19.

   Note:
   For the file <WPS_PATH>/PortalServer/zosinst/tools/deploy/client_env, make sure it is configured to set up the correct client environment for using the System Management Scripting API of WebSphere Application Server for z/OS or OS/390.

5. This job runs the shell script WPAConfig.sh which installs the portlets on the application server.
6. Check the job log for successful completion.
7. From the TSO console, start WebSphere Portal via the command:

   `/s <WPS_PROCNAME>.<WPS_SERVERNAME>A.`

   For example, we used: **`/s WSPORT.WSPORTA.`**

   The startup performs naming registration for newly-deployed portlets and displays console messages about starting and completing the registration. Wait until the naming registration has successfully completed before using the newly deployed portlets.

Portlet deployment using the WPAConfig script

The steps to deploy a portlet using WPAConfig script are:

1. Bring up the login portal page by entering the following URL in a Web browser: `http://<HOST_NAME:8082>/wps/myportal`.
2. Log on to WebSphere portal as a user with administrator rights and install, delete, or update portlets via the Portal Administration GUI. Make sure this step was successful.
3. Go to a z/OS or OS/390 console screen, stop WebSphere Portal via the command: **`/p <WPS_PROCNAME>.<WPS_SERVERNAME>A`**. For example: **`/p WSPORT.WSPORTA.`**
4. Invoke the WPAConfig.sh script to deploy the portlets to the hosting WebSphere Application Server to make them operational. To do this:
   a. Log on to the UNIX System Services on the target system as a user with administrator authority for WebSphere Application Server.
   b. Change directory to <WPS_PATH>/PortalServer/zosinst/tools/deploy
   c. Make sure the environment of this user is setup to use the SM Scripting API of WebSphere Application Server for z/OS or OS/390. If you need to perform this setup, run the batch file client_env, as follows:. client_env
   d. Run **`WPAConfig`**. This will install, delete, or update the portlet, as the case may be, on the WebSphere Application Server for z/OS or OS/390.
   e. Wait for the conversation committed message:

      ```
      Conversation WPSConversation_xxxxxx_hh:mm:ss was committed.
      ```
5. From the z/OS or OS/390 console, start WebSphere Portal via the command: **`/s <WPS_PROCNAME>.<WPS_SERVERNAME>A,srvname='<WPS_SERVERNAME>A'`**. For example: **`/s WSPORT.WSPORTA`**. This startup performs naming registration for newly-deployed portlets and displays console messages about starting and completing the registration.
6. Wait until the naming registration has successfully completed before using the portlets.

6.3.5 Verifying deployment

First check that the processing of the job and script on z/OS was successful. Check the log files for error messages or warnings. The totals are displayed at the end of the log file as messages WP00047I and WP00048I. If there are warnings and no error messages, everything was, most likely, successfully configured.

Jobs that have errors are moved to the failed directory and not to the done directory. Therefore, after the problem is fixed, move the job from the *failed* directory to the *jobs* or *work* directory. Then re-run the WAPCONF job or WPAConfig script.

If everything was successful, go to a Web browser and bring up the default portal page, via the URL `http://<HOST_NAME>:8082/wps/portal`. If need be, login and navigate to the page where the new portlets are placed. If the portlets were placed on a place and page after Step 1 of the deployment process then they should now be visible without errors.

6.4 Portlet deployment examples

In this section we show two examples of deploying portlets on z/OS as follows:

1. Using WPACONF job via TSO to deploy a portlet from the Portlet Catalog. Section 6.4.3, “Installing portlets obtained from the Portlet Catalog” on page 215 provides more information about portlets from the Portlet Catalog.
2. Using the WPAConfig script via USS to deploy an out of the box portlet installed during portal server installation but not deployed.

6.4.1 Example of installing a portlet using WPACONF job

As our first example of deploying a portlet, we used the *Health Care click-to-action* (C2A) sample portlet available for download from the Portlet Catalog (NAVCODE 1WP10003L).

Install the portlet via WebSphere Portal administration GUI

There are a number of different paths that an administrator can take to deploy portlets and make them available for users on a page. In fact, the only steps an administrator has to do is deploy the portlets and assign user rights so that users can add them to a page for themselves. In this example, we performed the following steps as an administrator to install the portlet:

1. Download the sample from the Portlet Catalog Web site to a Windows system and unzip the file. Follow the instructions in the supplied documentation in the form of file `readme.html`.

Tip: Because of the installation differences between distributed platforms and z/OS we used the following directories on z/OS instead of the ones referred to in the portlet installation instructions:

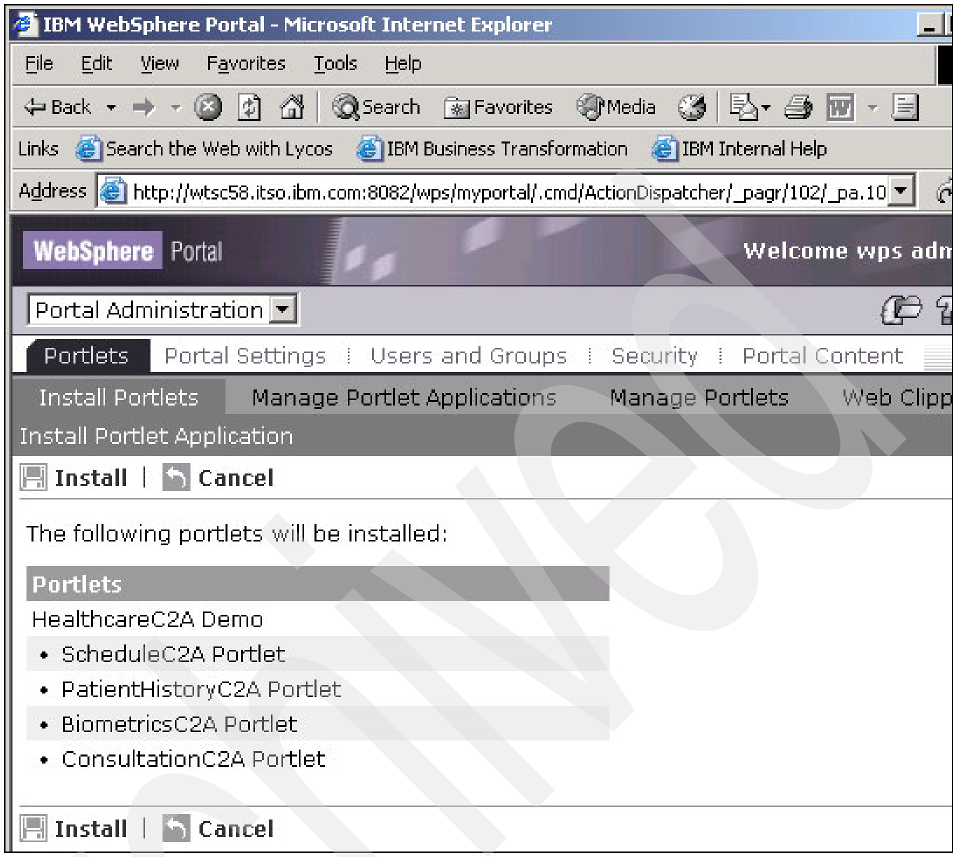
- Step 3: Used directory <WPS_PATH>/PortalServer/libbapp instead of <was_root>\lib\app.
- Step 4: Used directory <WPS_PATH>/PortalServer/libbapp/nls instead of <was_root>\lib\app\nls.
- Step 5: Created directory c2a in directory <WS_CONFIG_PATH>/apps/WSPORT/WPS Enterprise Application\wps.war instead of <wp_root>\app\wps.ear\wps.war.
- Step 6: Used directory <WPS_PATH>/libapp/WEB-INF/tld instead of <was_root>\lib\applWEB-INF\tld.

Tip: We recommend that changes are made, if required, to `properties` and `xml` files before deploying portlets on z/OS. This is because some of these files are in ASCII format, and some are stored inside the runtime of the expanded WAR file on the application server, making it difficult to change them after deployment.

2. Stop the Portal Server from a TSO session and then restart it.

   Note: The following steps will fail for the C2A portlets, if the Portal server is not stopped and restarted after following the portlet installation Steps 3-8 in the Readme.

3. Ensure that WebSphere Portal server on z/OS is running, and from a Web browser go to the following URL: `http://<HOST_NAME>/wps/portal`.
4. Login as the portal administrator with the default userid and password of *wpsadmin* as shown in Figure 6-17 on page 208.
5. Go to the **Portal Administration** place and select **Install Portlets**.
6. Browse the directory on the distributed system that corresponds to <WPS_PATH>/PortalServer/app/DeployablePortlets directory and select `healthcarec2a.war`. Click **Next**.
7. On the Install Portlet Application screen, the portlet(s) contained in the WAR and to be deployed are shown as in Figure 6-13 on page 203.
8. Click **Install** to install all the C2A Health Care sample portlets and ensure you receive the message `Portlets successfully installed.`


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://wtsc58.itso.ibm.com:8082/wps/myportal/.cmd/ActionDispatcher/_pagr/102/_pa.10
WebSphere Portal
Welcome wps adm
Portal Administration
Portlets Portal Settings Users and Groups Security Portal Content
Install Portlets Manage Portlet Applications Manage Portlets Web Clipp
Install Portlet Application
Install | Cancel

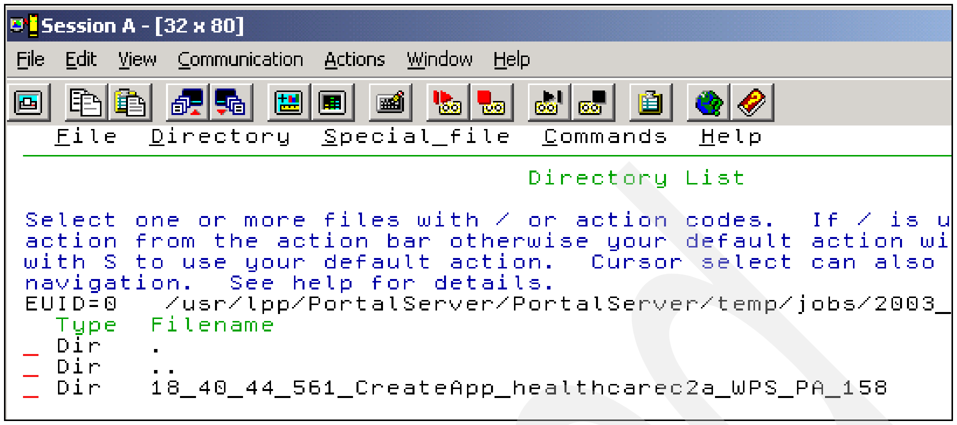
The following portlets will be installed:

Portlets
HealthcareC2A Demo
• ScheduleC2A Portlet
• PatientHistoryC2A Portlet
• BiometricsC2A Portlet
• ConsultationC2A Portlet

Figure 6-13 Installing C2A portlets from the Portal Administration place

Run the WPACONF job

Logon to TSO and using the TSO ISH environment you can optionally verify that the portlet deployment job(s) have been added to directory <WPS_PATH>/temp/jobs as shown in Figure 6-14 on page 204.



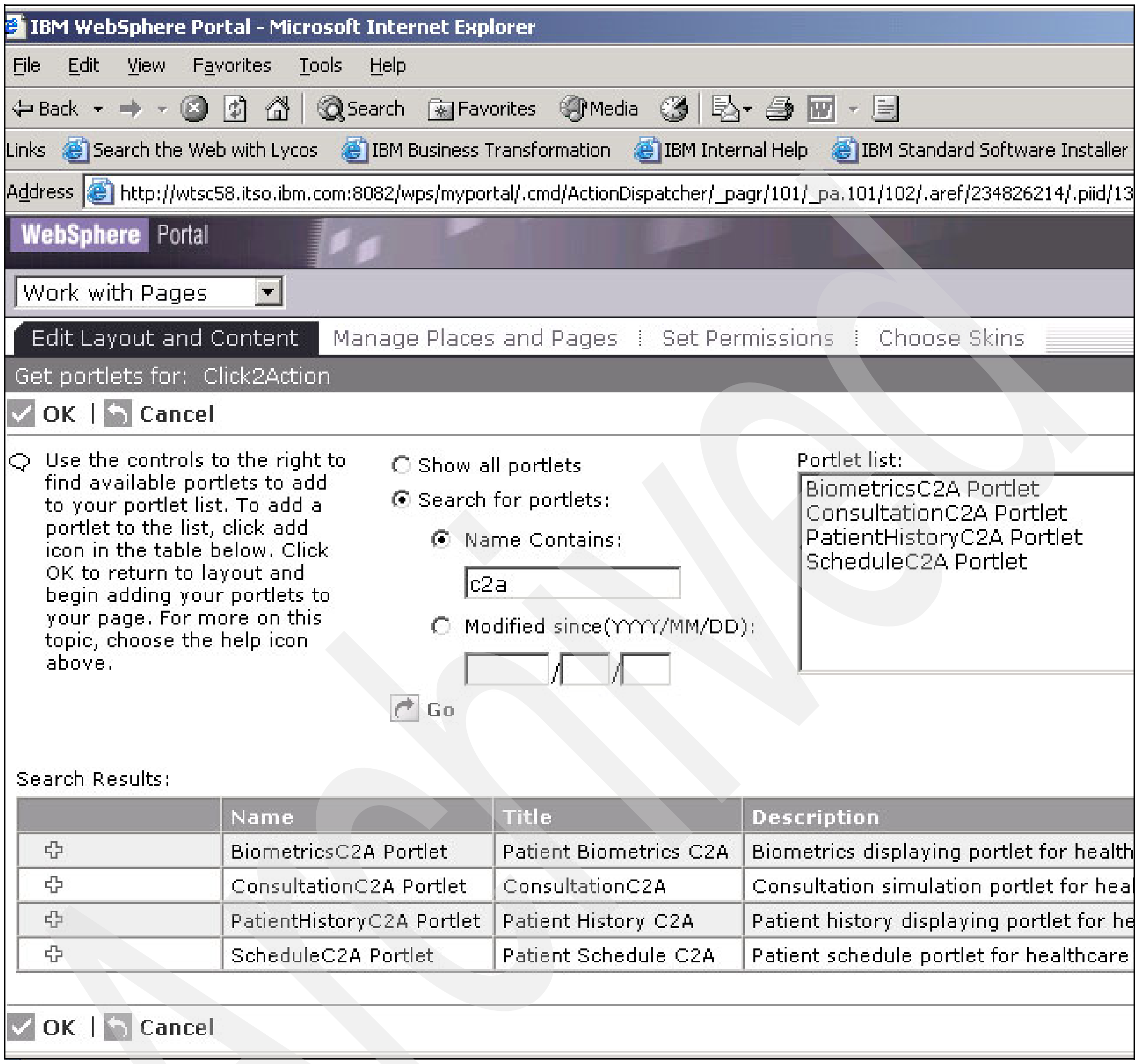
```
   File  Directory  Special_file  Commands  Help
 ──────────────────────────────────────────────────────────
                                Directory List

 Select one or more files with / or action codes.  If / is u
 action from the action bar otherwise your default action wi
 with S to use your default action.  Cursor select can also
 navigation.  See help for details.
 EUID=0    /usr/lpp/PortalServer/PortalServer/temp/jobs/2003_
   Type   Filename
 _ Dir    .
 _ Dir    ..
 _ Dir    18_40_44_561_CreateApp_healthcarec2a_WPS_PA_158
```

Figure 6-14 C2A deploy work ready to run job WPACONF

Submit job WPACONF, after reviewing the JCL job card. Note that if the Portal Server is running it will be stopped and re-started for the deployment of each portlet it deploys from the WAR files in the work directory. Therefore, we recommend stopping Portal Server before submitting the job and re-starting it manually on completion. Check the job completes without errors.

Add the portlet to a page

Start Portal Server and login using the administrative user ID via `http://<yourhost>/wps/portal`. Navigate to the page you want to add the newly deployed portlets via Work with pages -> Edit layout and content -> Place+Page -> Get Portlets. We chose a filtered search on C2A and selected the Health Care sample portlets to be added to our page as shown in Figure 6-15 on page 205.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installer
Address http://wtsc58.itso.ibm.com:8082/wps/myportal/.cmd/ActionDispatcher/_pagr/101/_pa.101/102/.aref/234826214/.piid/13
WebSphere Portal
Work with Pages
Edit Layout and Content | Manage Places and Pages | Set Permissions | Choose Skins
Get portlets for: Click2Action
OK | Cancel

Use the controls to the right to find available portlets to add to your portlet list. To add a portlet to the list, click add icon in the table below. Click OK to return to layout and begin adding your portlets to your page. For more on this topic, choose the help icon above.

( ) Show all portlets
(●) Search for portlets:
(●) Name Contains: c2a
( ) Modified since(YYYY/MM/DD): / /
Go

Portlet list:
BiometricsC2A Portlet
ConsultationC2A Portlet
PatientHistoryC2A Portlet
ScheduleC2A Portlet

Search Results:

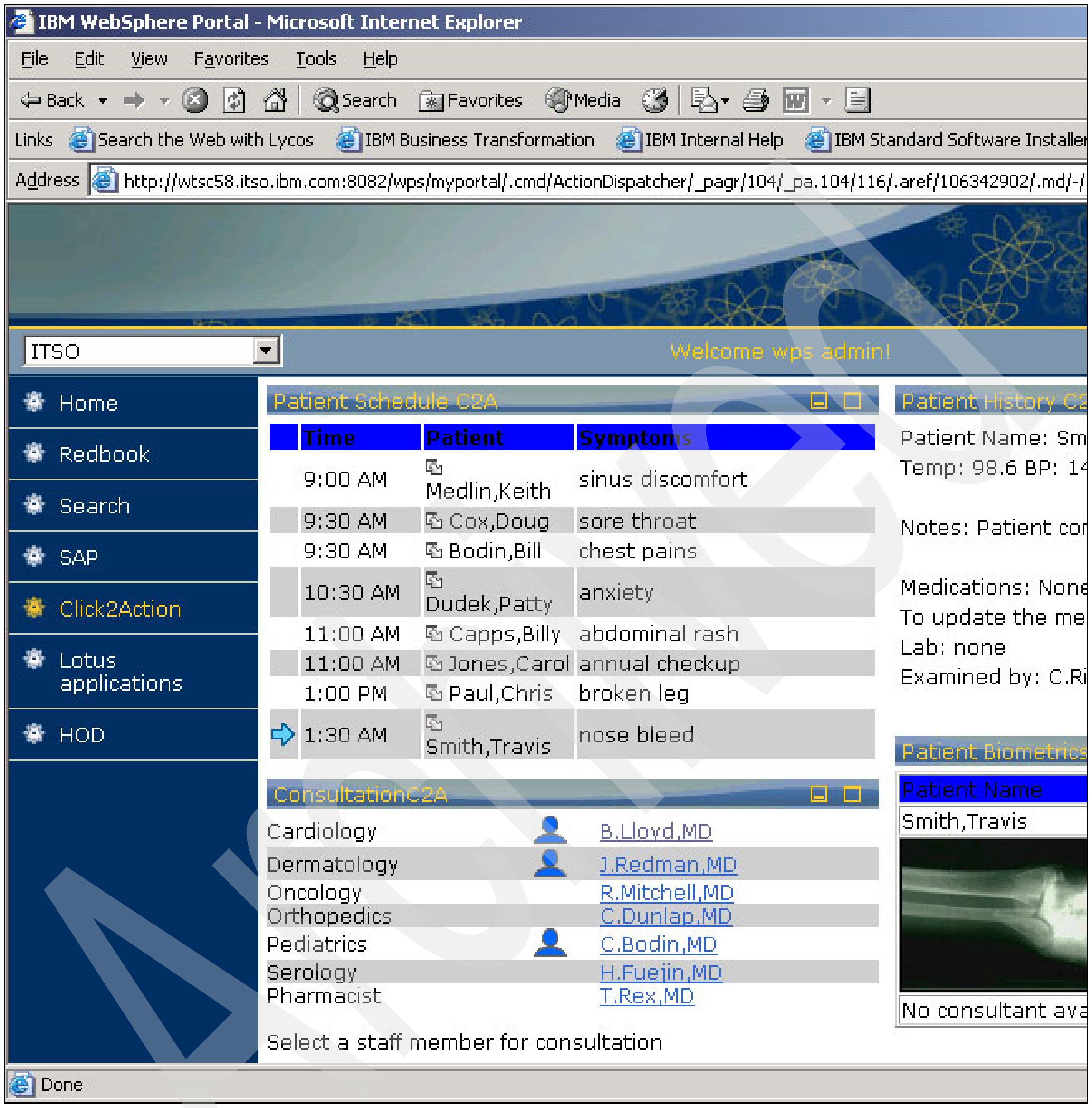
	Name	Title	Description
✛	BiometricsC2A Portlet	Patient Biometrics C2A	Biometrics displaying portlet for health
✛	ConsultationC2A Portlet	ConsultationC2A	Consultation simulation portlet for hea
✛	PatientHistoryC2A Portlet	Patient History C2A	Patient history displaying portlet for he
✛	ScheduleC2A Portlet	Patient Schedule C2A	Patient schedule portlet for healthcare

Figure 6-15 Adding the C2A portlets to a page

The next portal administration screen allows you to select the row and column containers to show the portlets on the selected page. Finally, ensure that page is activated.

Verify that the portlet works

Navigate to the page and the deployed portlets should be available and working, as shown for example in Figure 6-16 on page 206.


IBM WebSphere Portal - Microsoft Internet Explorer

Address http://wtsc58.itso.ibm.com:8082/wps/myportal/.cmd/ActionDispatcher/_pagr/104/_pa.104/116/.aref/106342902/.md/-/

Welcome wps admin!

Patient Schedule C2A

	Time	Patient	Symptoms
	9:00 AM	Medlin,Keith	sinus discomfort
	9:30 AM	Cox,Doug	sore throat
	9:30 AM	Bodin,Bill	chest pains
	10:30 AM	Dudek,Patty	anxiety
	11:00 AM	Capps,Billy	abdominal rash
	11:00 AM	Jones,Carol	annual checkup
	1:00 PM	Paul,Chris	broken leg
⇨	1:30 AM	Smith,Travis	nose bleed

ConsultationC2A

Cardiology	B.Lloyd,MD
Dermatology	J.Redman,MD
Oncology	R.Mitchell,MD
Orthopedics	C.Dunlap,MD
Pediatrics	C.Bodin,MD
Serology	H.Fuejin,MD
Pharmacist	T.Rex,MD

Select a staff member for consultation

Patient History C2

Patient Name: Sm
Temp: 98.6 BP: 14

Notes: Patient co

Medications: None
To update the me
Lab: none
Examined by: C.Ri

Patient Biometrics

Patient Name
Smith,Travis

No consultant ava

Done

Figure 6-16 C2A example portlets

6.4.2 Example of installing a portlet using WPAConfig

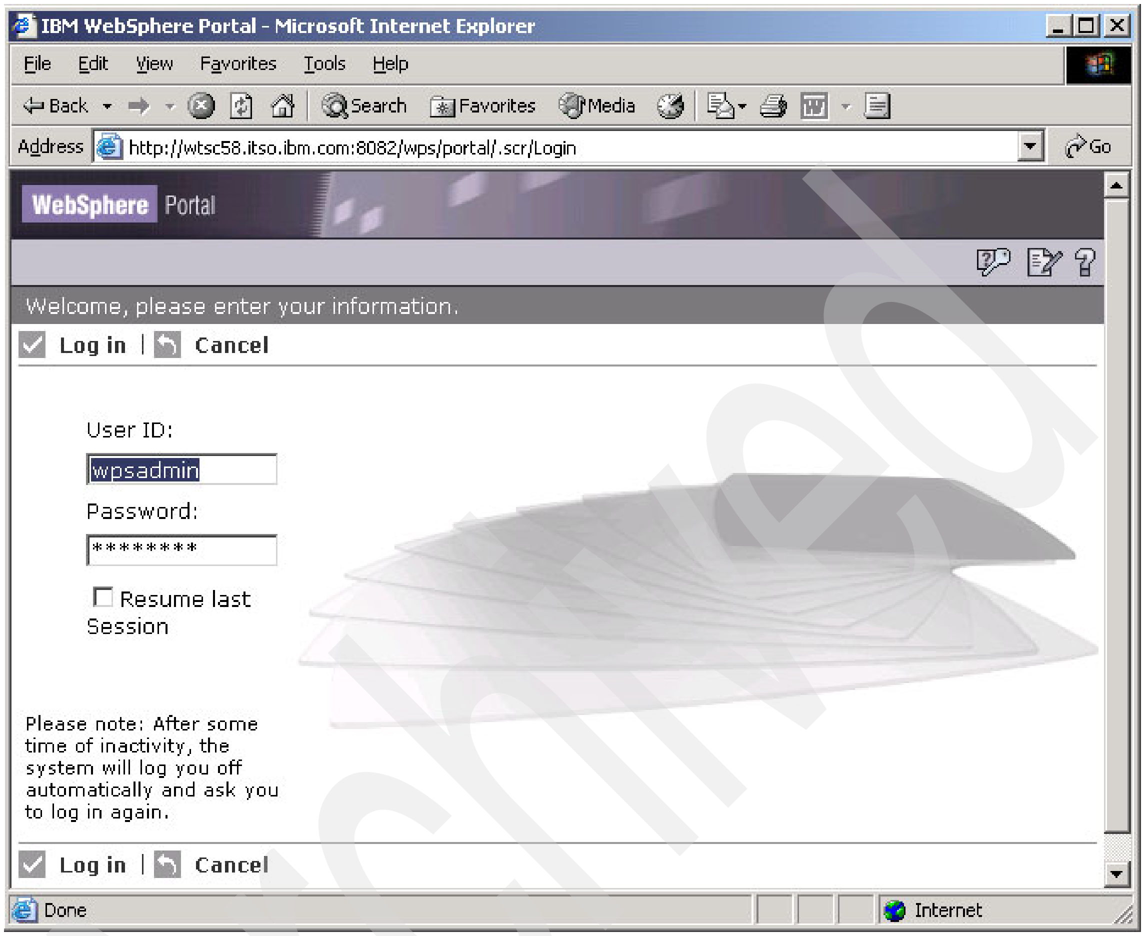
In this section we take you through the steps of installing a portlet supplied with WebSphere Portal but not installed as part of the post installation process. All portlets that come with WebSphere Portal can be found as WAR files in

<WPS_PATH>/PortalServer/app/DeployablePortlets directory. We will install the Document Viewer portlet and place it on a page we created called Redbook that exists in the ITSO place that we also created. The following steps are enumerated:

Tip: Since the HFS on z/OS cannot be accessed directly from a browser or as a mapped drive, copy the portlet WAR file(s) to a distributed system where you are running the Web browser.

Install the portlet via WebSphere Portal administration GUI

1. Assuming that WebSphere Portal server on z/OS is running, bring up a Web browser and go to the default URL: <HOST_NAME>/wps/myportal.
2. Login as the portal administrator with the default userid and password of *wpsadmin* as shown in Figure 6-17 on page 208.


IBM WebSphere Portal - Microsoft Internet Explorer

Address: http://wtsc58.itso.ibm.com:8082/wps/portal/.scr/Login

Welcome, please enter your information.

Log in | Cancel

User ID: wpsadmin

Password: ********

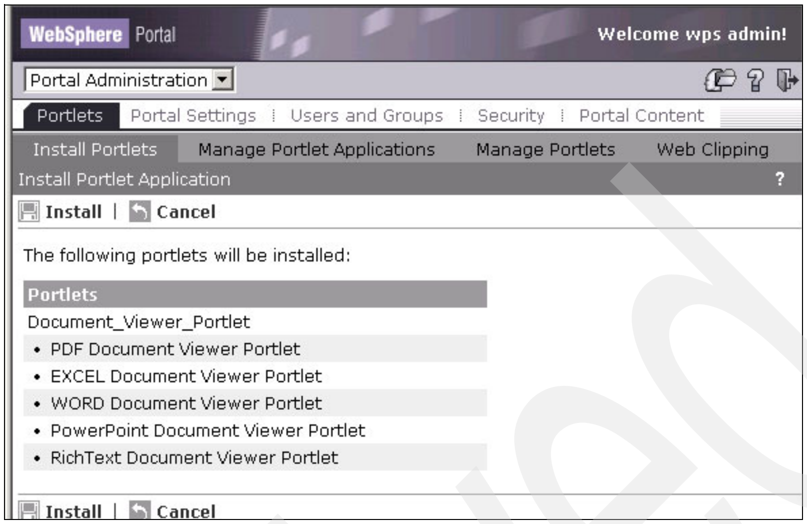
☐ Resume last Session

Please note: After some time of inactivity, the system will log you off automatically and ask you to log in again.

Log in | Cancel

Figure 6-17 WebSphere Portal login window

3. Go to **Portal Administration Pages** and make sure the **Install Portlets** page is selected.
4. Browse to the directory on the workstation where you transferred docviewer.war from the z/OS HFS directory <WPS_PATH>/PortalServer/app/DeployablePortlets and select **docviewer.war**. See Figure Figure 6-18 on page 209. Click **Next**.

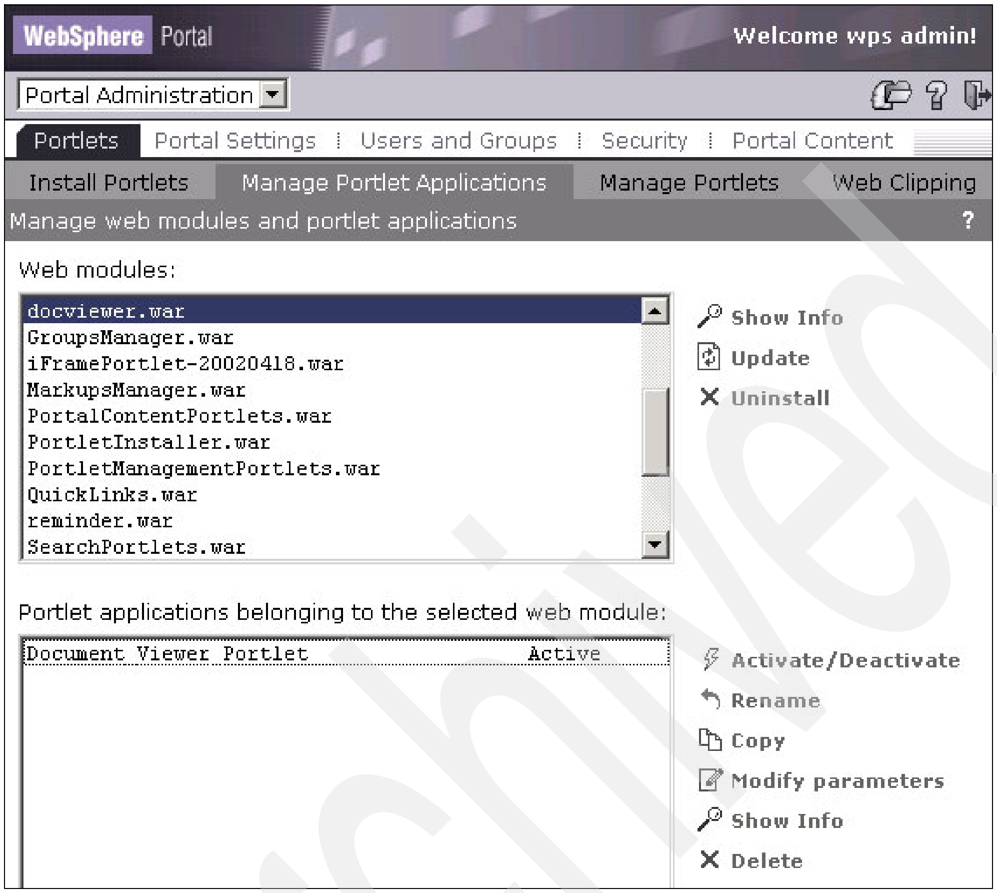


The following portlets will be installed:

Portlets
Document_Viewer_Portlet
• PDF Document Viewer Portlet
• EXCEL Document Viewer Portlet
• WORD Document Viewer Portlet
• PowerPoint Document Viewer Portlet
• RichText Document Viewer Portlet

Figure 6-18 Initial window on portlet installation

5. On the **`Install Portlet Application`** page, you will the portlet(s) contained in the WAR file. Notice that the document viewer portlet is a portlet application that contains 5 separate portlets.
6. Click **`Install`** to install all the Document Viewer Portlets. This will take a few moments.
7. Wait until you receive the message - `Portlets successfully installed`.
8. If you go to the **`Manage Portlet Applications`** tab, and look for the `docviewer.war` in the list of Web modules, you will see that the Portlet application belonging to it are `Active` as seen in Figure Figure 6-19 on page 210.



Web modules:

```
docviewer.war
GroupsManager.war
iFramePortlet-20020418.war
MarkupsManager.war
PortalContentPortlets.war
PortletInstaller.war
PortletManagementPortlets.war
QuickLinks.war
reminder.war
SearchPortlets.war
```

Portlet applications belonging to the selected web module:

```
Document Viewer Portlet                    Active
```

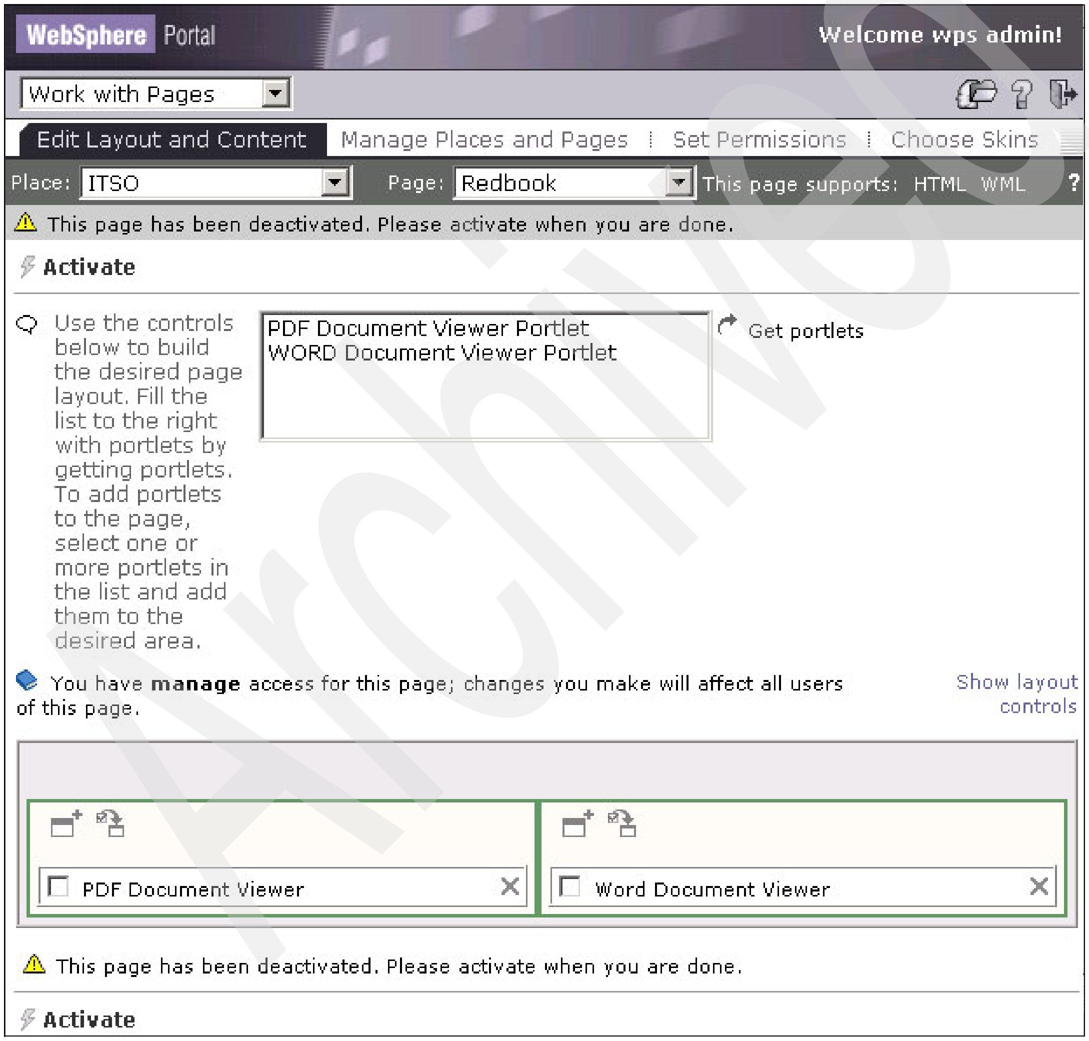
Figure 6-19 Manage Portlet Applications window

Place the portlet on a portal page

1. Select **Work with Pages** place and click on **Edit Layout and Content**.
2. Select **ITSO** in the Place drop down and select **Redbook** in the Page drop down list.
3. Click **Get portlets**.
4. On the following screen entitled **Get portlet for: ITSO**, click the radio button to **Show all portlets**. Click **Go**.
5. You should see a list of all the available portlets. We will install 2 of the 5 portlets that were contained in the docviewer WAR file. Click the plus icon on the left of **Name** field to add **PDF Document Viewer** and **Word Document Viewer**. Those portlets will be added to the Portlet list. Click **OK** in the portlet menu bar.

6. You will be back at the **`Edit layout and Content`** window for the Redbook page. Highlight **`PDF Document Viewer Portlet,`** move your cursor to the left side of one of the columns on the page and click the **`Add`** icon.

   Then click to select the **`WORD Document Viewer Portlet`** and click the **`Add portlet`** icon in the right-side column. The resulting screen is shown Figure 6-20 on page 211.



This page has been deactivated. Please activate when you are done.

Use the controls below to build the desired page layout. Fill the list to the right with portlets by getting portlets. To add portlets to the page, select one or more portlets in the list and add them to the desired area.

PDF Document Viewer Portlet
WORD Document Viewer Portlet

You have **manage** access for this page; changes you make will affect all users of this page.

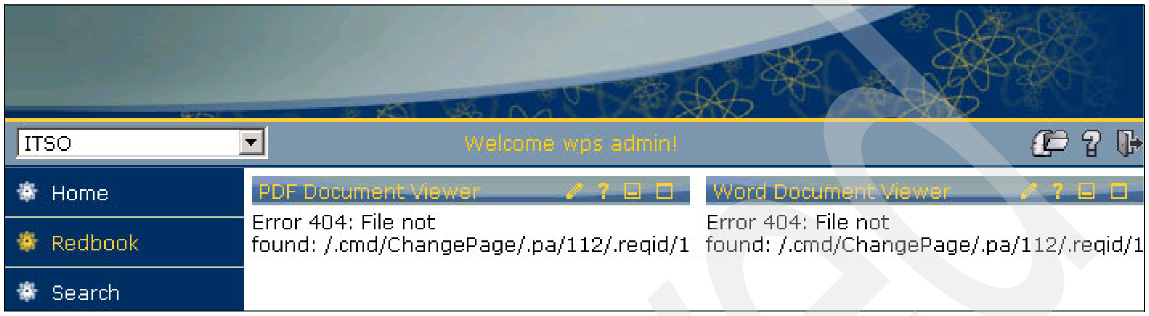
PDF Document Viewer

Word Document Viewer

This page has been deactivated. Please activate when you are done.

Figure 6-20 Laying out the portlets in portlet containers on a page

7. Remember to click the **`Activate`** label so that the page is activated and can be viewed.
8. From the place drop down list, select **`ITSO`**, navigate to the **`Redbook`** page and you should see the portlet outlines with a Error 404: File not found message as seen in Figure 6-21 on page 212.



Error 404: File not found: /.cmd/ChangePage/.pa/112/.reqid/1

Error 404: File not found: /.cmd/ChangePage/.pa/112/.reqid/1

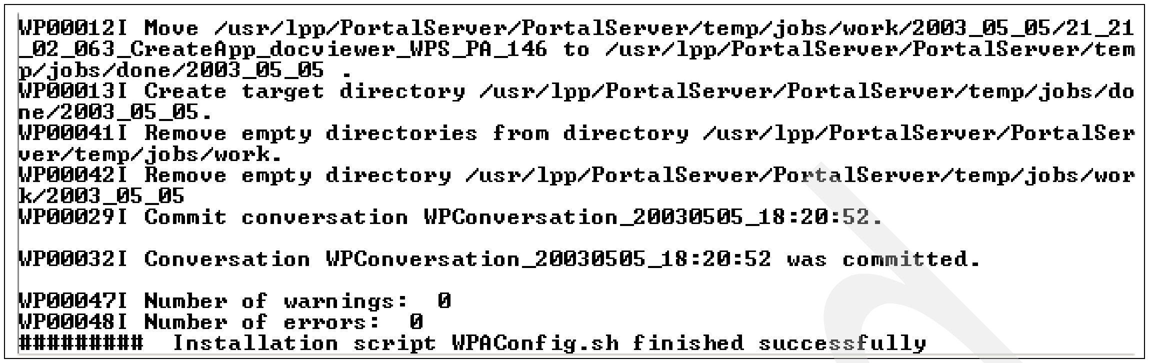
Figure 6-21 Screen showing the newly installed portlets with the error message

Run the WPAConfig script

1. Go to a z/OS TSO session screen and stop WebSphere Portal, in our case we issued command: **`/p WSPORTA`**.
2. Log on to the UNIX System Services on the target system as a user with administrator authority for WebSphere Application Server. (On our system we logged in as `CBADMIN`.)
3. Change directory to

   `/usr/lpp/PortalServer/PortalServer/zosinst/tools/deploy.`
4. If need be run the batch file client_env, as follows:. **`client_env`**
5. Change directory to /usr/lpp/PortalServer/PortalServer/zosinst/scripts. Invoke the **`WPAConfig.sh`** command. This will run the job, start a conversation, and install the portlets on the WebSphere Application Server for z/OS and OS/390.
6. Wait for the committed message (Figure 6-22 on page 213):

   ```
   Conversation WPSConversation_xxxxxxxx_hh:mm:ss was committed.
   ```



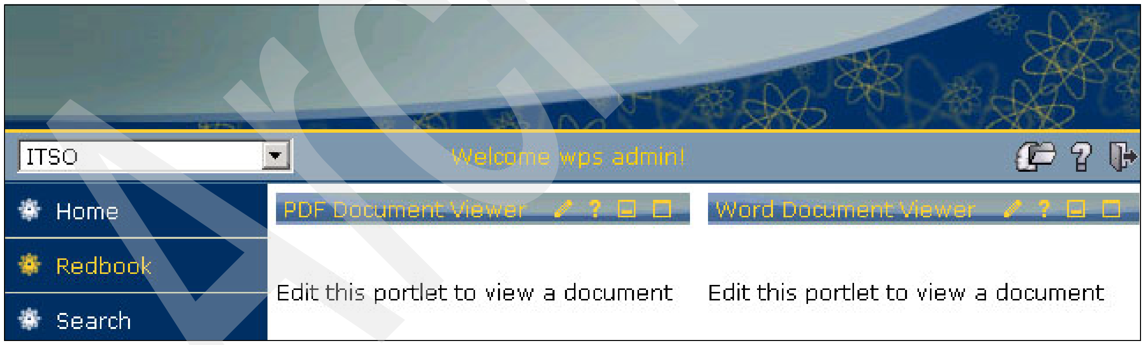
```
WP00012I Move /usr/lpp/PortalServer/PortalServer/temp/jobs/work/2003_05_05/21_21
_02_063_CreateApp_docviewer_WPS_PA_146 to /usr/lpp/PortalServer/PortalServer/tem
p/jobs/done/2003_05_05 .
WP00013I Create target directory /usr/lpp/PortalServer/PortalServer/temp/jobs/do
ne/2003_05_05.
WP00041I Remove empty directories from directory /usr/lpp/PortalServer/PortalSer
ver/temp/jobs/work.
WP00042I Remove empty directory /usr/lpp/PortalServer/PortalServer/temp/jobs/wor
k/2003_05_05
WP00029I Commit conversation WPConversation_20030505_18:20:52.

WP00032I Conversation WPConversation_20030505_18:20:52 was committed.

WP00047I Number of warnings:  0
WP00048I Number of errors:  0
#########  Installation script WPAConfig.sh finished successfully
```

Figure 6-22 Command window output of running the WPAConfig script

7. Make sure there are no error messages. At this stage the portlets are loaded and portal server needs to be restarted.
8. From the z/OS console, start WebSphere Portal, in our case we issued command: `/s WSPORT.WSPORTA`.

Verify that the portlet works

1. Refresh the browser that was displaying the Redbook page in the ITSO place or bring up a new browser and go to the ITSO place and down to the Redbook page. The two portlets should be displayed without any errors as shown in Figure 6-23 on page 213.

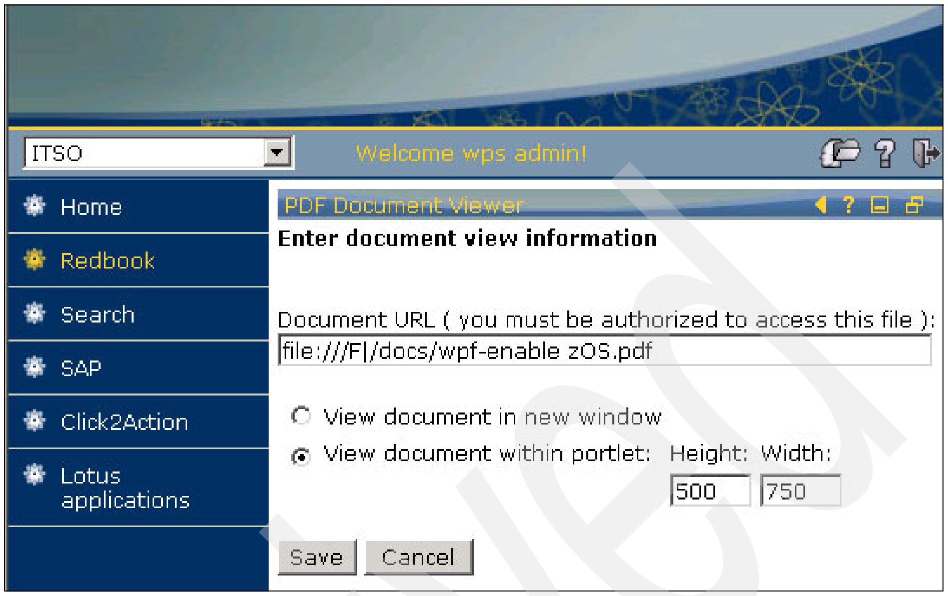


Edit this portlet to view a document

Edit this portlet to view a document

Figure 6-23 Screen showing the docviewer portlets before configuration

2. Click the **Edit** icon (one that looks like a pencil) in the menu bar at the top of one of the portlets. Enter the name of the PDF document you want to view. You have the choice of displaying the document in a new window or within the portlet as shown in Figure 6-24 on page 214. After all the parameters are entered, click **Save**.



Enter document view information

Document URL (you must be authorized to access this file): file:///F|/docs/wpf-enable zOS.pdf

( ) View document in new window
[✓ View document within portlet:] Height: 500 Width: 750

Figure 6-24 Configuring the PDF document viewer portlet with document details

3. You will taken back to the Redbook page and the contents of the PDF will be displayed.
4. You can similarly deploy and use the other portlets in the docview Web module.

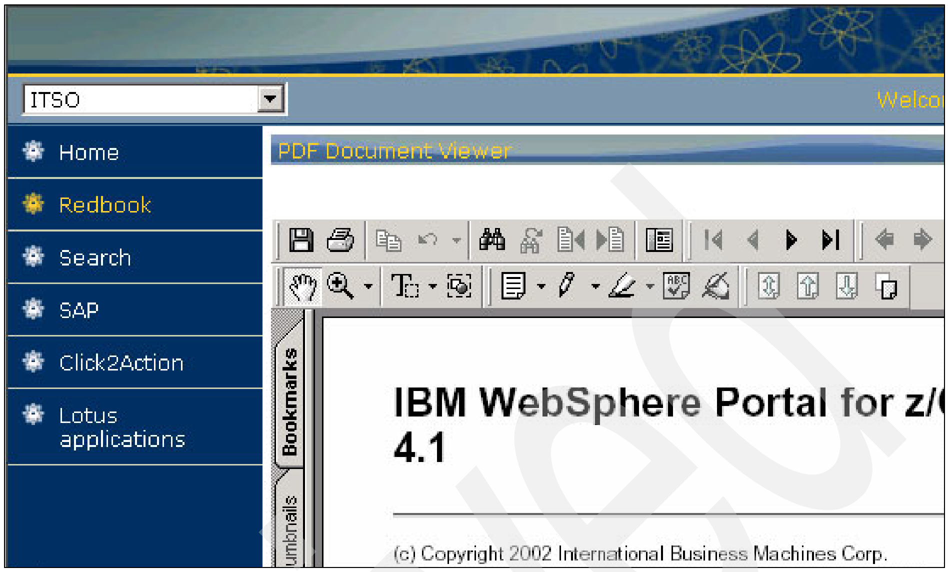


Figure 6-25 PDF file viewed in the portlet

6.4.3 Installing portlets obtained from the Portlet Catalog

There are hundreds of portlets in the IBM Portlet Catalog to be found at

http://www-3.ibm.com/services/cwi/portal/_pagr/105/

Many of them are available for download for free although some provided by third parties require registration and a fee.

Note: Portlets are categorized in the IBM Portlet Catalog for different versions of WebSphere Portal and different editions. We recommend ensuring you download portlets for *WebSphere Portal Enable V4.1* or *V4.2*.

In the following examples the installation steps are the same as we have seen in the preceding sections on portlet deployment. For each of the examples we describe any unique configuration steps that are performed.

Host on Demand (HOD) portlet

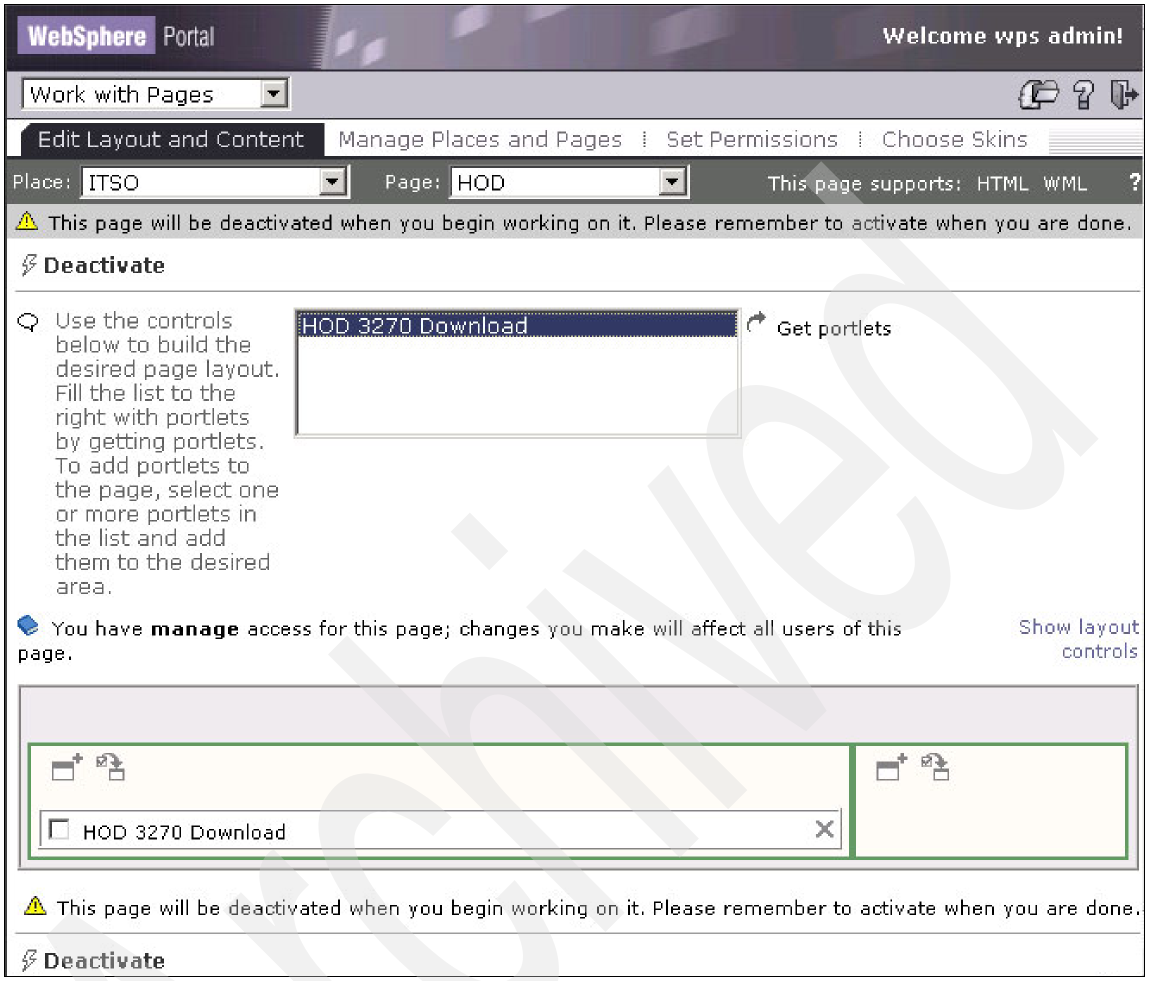
In this example we will download the WebSphere Host On Demand (HOD) portlet, install and deploy it. There are 4 HOD portlets in the catalog - HOD 3270 cached and non-cached and HOD 5250 Cached and Non-cached. We will use the HOD 3270 Cached portlet for WebSphere Portal V4.1.4 (NavCode 1WP10006L).

Once the portlet WAR file has been downloaded the steps are same as listed in 6.4.2, “Example of installing a portlet using WPAConfig” on page 206 except for the fact that you browse the directory where the portlet WAR file was downloaded. We created a page called `HOD` in our portal `ITSO` place to place the 3270 Cached HOD portlet.

Assuming you are logged in as the portal administrator (or anyone else who has the authority to install portlets) do the following:

- Navigate to **`Work with Pages`**->**`Edit Layout and Content`**.
- Choose **`ITSO`** place and **`HOD`** page.
- Get and select the **`HOD 3270 Download`** portlet.
- Add it to a portlet container and **`Activate`** the page.

The portlet addition screen is shown in Figure 6-26 on page 217.


WebSphere Portal

Welcome wps admin!

Work with Pages

Edit Layout and Content | Manage Places and Pages | Set Permissions | Choose Skins

Place: ITSO Page: HOD This page supports: HTML WML ?

This page will be deactivated when you begin working on it. Please remember to activate when you are done.

Deactivate

Use the controls below to build the desired page layout. Fill the list to the right with portlets by getting portlets. To add portlets to the page, select one or more portlets in the list and add them to the desired area.

HOD 3270 Download

Get portlets

You have **manage** access for this page; changes you make will affect all users of this page.

Show layout controls

HOD 3270 Download

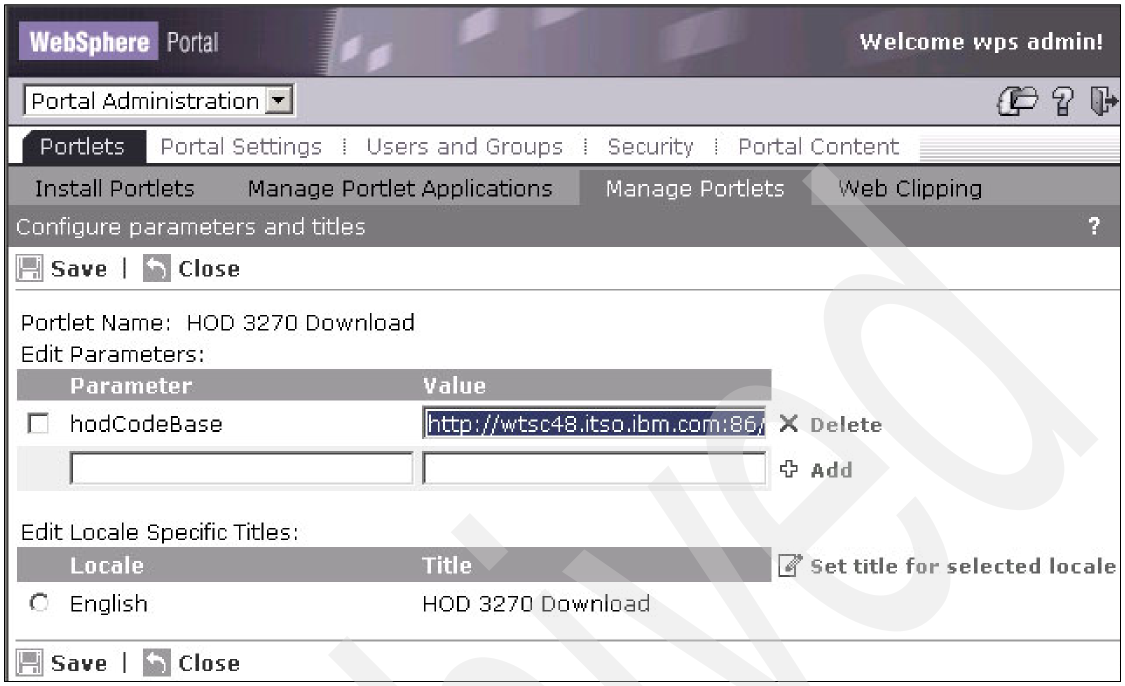
This page will be deactivated when you begin working on it. Please remember to activate when you are done.

Deactivate

Figure 6-26 Window showing the placement of HOD portlet on the HOD page

The HOD portlet has to be configured like this, before you can actually work with it:

- Navigate to **`Portal Administration`** -> **`Portlets`** -> **`Manage Portlets`**.
- In the list of Portlets, highlight **`HOD 3270 Download`** and select **`Modify parameters.`**
- In the **`hodCodeBase`** parameter enter the name of the server that is running the HOD server and point to the hod directory. In our example we used `http://wtsc48.itso.ibm.com:86/hod`.
- Click **`Save`**. Then click **`Close`**.



Portlet Name: HOD 3270 Download

Edit Parameters:

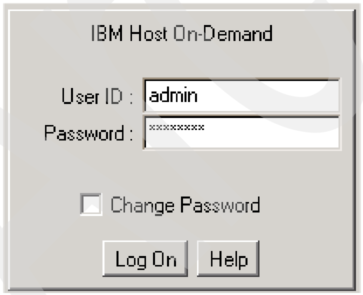
Parameter	Value
hodCodeBase	http://wtsc48.itso.ibm.com:86/

Edit Locale Specific Titles:

Locale	Title
English	HOD 3270 Download

Figure 6-27 Screen to modify the parameters for the HOD portlet

Navigate to the ITSO place and bring up the page that has the HOD portlet, you should see the HOD portlet displaying a login screen as seen in Figure 6-28 on page 218. Enter the User ID and password and click **Log On**.



User ID : admin

Password : ********

Figure 6-28 HOD portlet login window

If there are any sessions configured, you should see icons reflecting them on the next screen. If you double-click on any icon then that session will be activated. A new floating screen will be displayed and an icon will appear in the Active Sessions pane of the HOD portlet. See Figure 6-29 on page 219. You can use this session like a normal 3270 session.

This ends the deployment and verification of the HOD portlet.

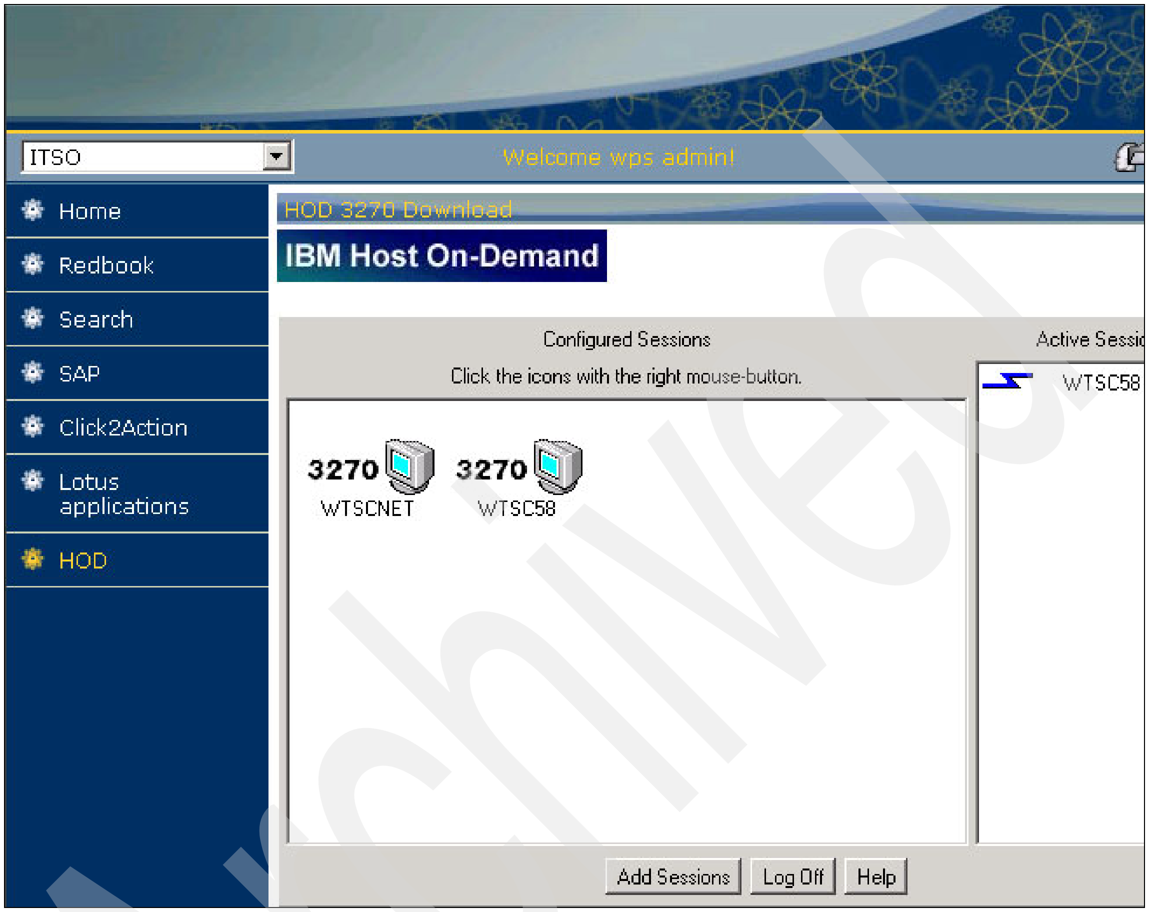


Figure 6-29 HOD portlet showing an active 3270 session

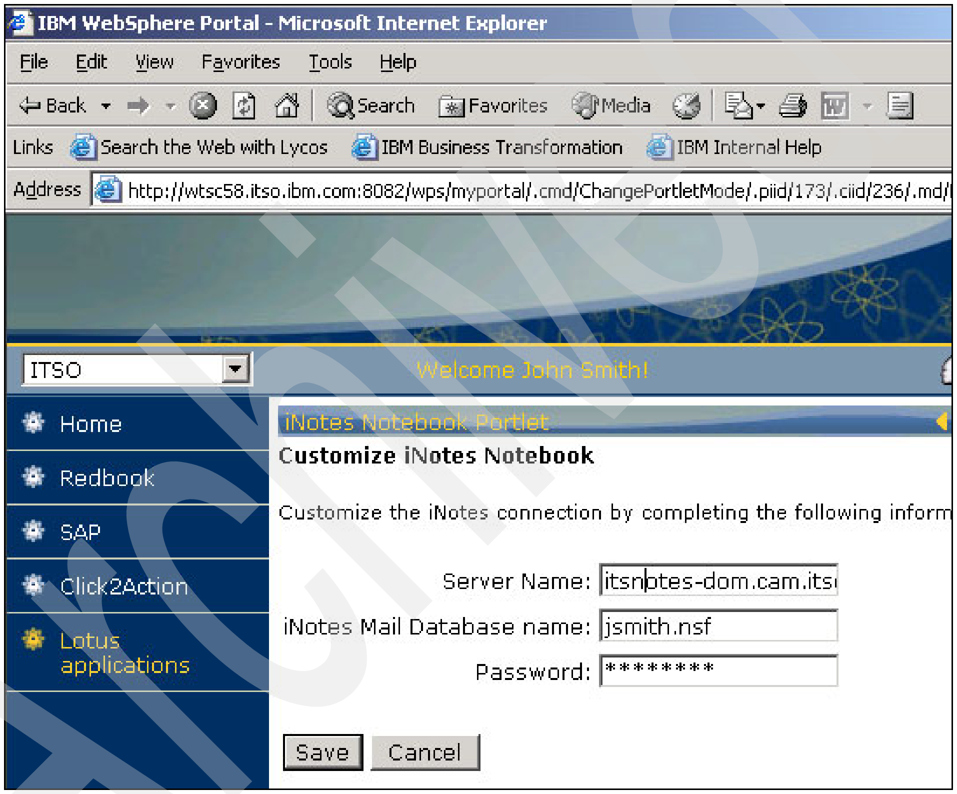
Lotus iNotes Portlets

The Portlet Catalog provides several portlets for accessing Lotus Notes® functions available on a Lotus Domino server, for example there are portlets called **iNotes**, Notes and MyNotes. The iNotes portlet can be used with the WebSphere Portal Enable edition giving users a browser view of the following Lotus Notes databases: mail, calendar, to-do, contacts and notebook.

We downloaded the Lotus iNotes Mail portlet from the Portlet Catalog NavCode: 1WP10007P which is saved as file inotes.war. The WAR file contains separate portlets for portlets mail, calendar, to-do, contacts and notebook and has a single configuration parameter authMethod for user sign on kept in the **portlet.xml** file in the WEB-INF directory inside the WAR file.

Tip: We recommend that changes are made if required to `web.xml` and **`portlet.xml`** before deploying portlets on z/OS. This is because these files are in ASCII format and stored inside the runtime of the expanded WAR file on the application server, making it difficult to change them after deployment.

After deploying the portlets, we clicked on the portlet edit icon, and then added the Notes server and user configuration properties to the iNotes portlets as shown in Notebook portlet example in Figure 6-30.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address
ITSO
Welcome John Smith!
Home
Redbook
SAP
Click2Action
Lotus applications
iNotes Notebook Portlet
Save
Cancel

http://wtsc58.itso.ibm.com:8082/wps/myportal/.cmd/ChangePortletMode/.piid/173/.ciid/236/.md/

Customize iNotes Notebook

Customize the iNotes connection by completing the following inform

Server Name: itsnotes-dom.cam.its

iNotes Mail Database name: jsmith.nsf

Password: ********

Figure 6-30 Edit the iNotes portlet

Click **Save** and the portlets will try to access the server and if successful you should see the Notes database as shown in the Mail portlet example in Figure 6-31 on page 221.

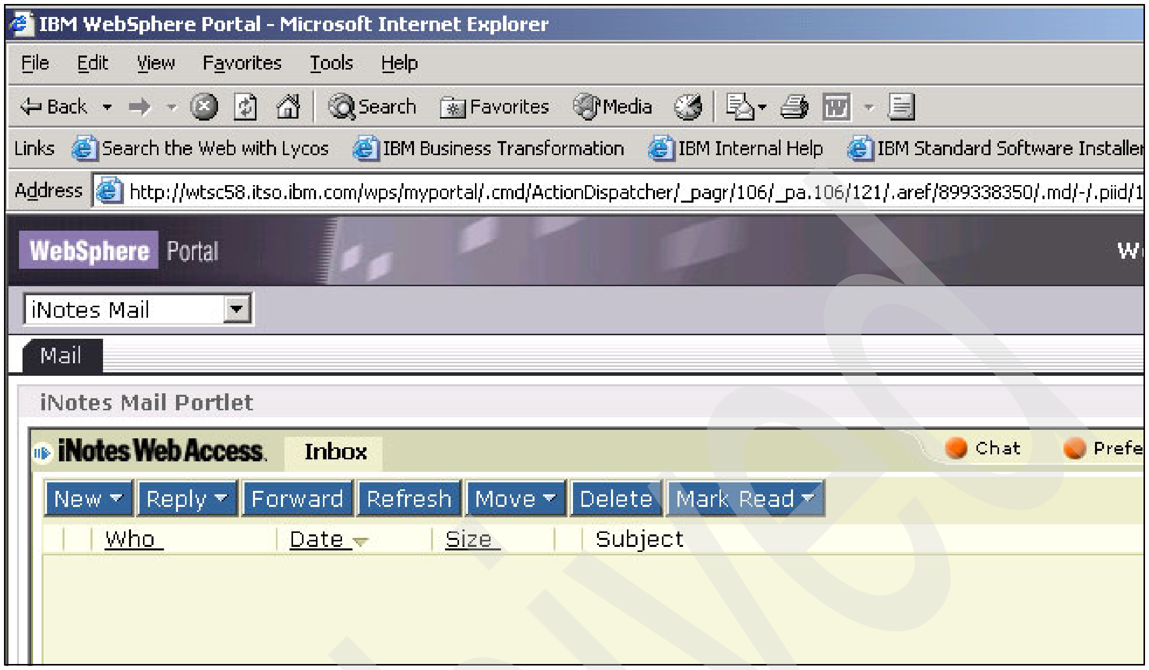
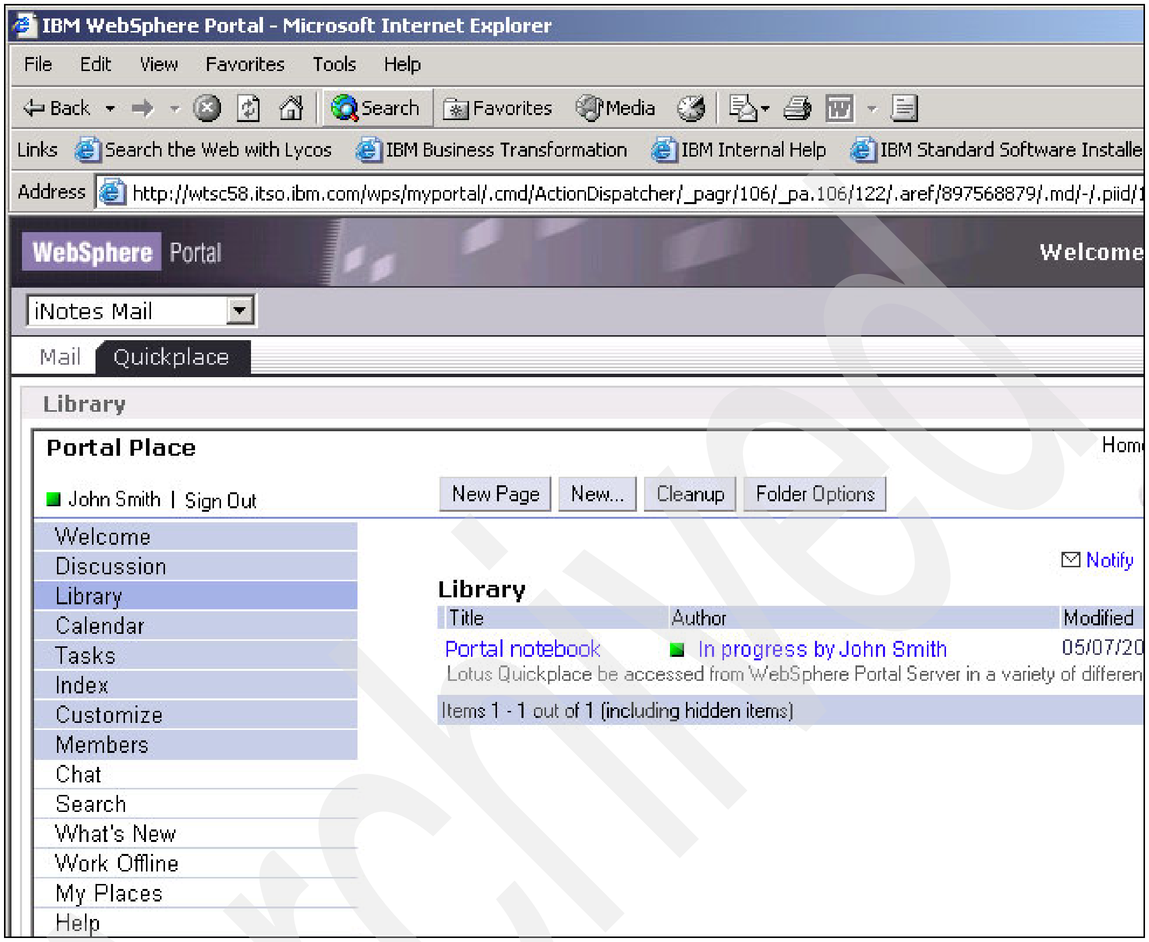

IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installer
Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispatcher/_pagr/106/_pa.106/121/.aref/899338350/.md/-/.piid/1
WebSphere Portal
iNotes Mail
Mail
iNotes Mail Portlet
iNotes Web Access Inbox
Chat Prefe
New Reply Forward Refresh Move Delete Mark Read
Who Date Size Subject

Figure 6-31 iNotes Mail portlet

Lotus Quickplace

WebSphere Portal on z/OS can also be used to integrate access to Lotus Quickplace and there are portlets for this available in the Portlet Catalog. We cheated by using the `iFrame` portlet that was already installed on the portal server to access our Quickplace server with quite good results as shown in “Accessing Lotus Quickplace from Portal Server” on page 222.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installe
Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispatcher/_pagr/106/_pa.106/122/.aref/897568879/.md/-/.piid/1
WebSphere Portal
Welcome
iNotes Mail
Mail
Quickplace
Library
Portal Place
Hom
John Smith | Sign Out
New Page
New...
Cleanup
Folder Options
Welcome
Discussion
Library
Calendar
Tasks
Index
Customize
Members
Chat
Search
What's New
Work Offline
My Places
Help
Notify
Library

Title	Author	Modified
Portal notebook	🟩 In progress by John Smith	05/07/20

Lotus Quickplace be accessed from WebSphere Portal Server in a variety of differen

Items 1 - 1 out of 1 (including hidden items)

Figure 6-32 Accessing Lotus Quickplace from Portal Server

6.4.4 Deploy the Hello World Portlet

In 6.1.1, “Using Application Developer to develop a portlet” on page 184 we developed a portlet called HelloWorld.

The steps for installing and deploying a custom portlet developed using Application Developer are the same as listed in 6.3.1, “Portlet deployment jobs” on page 195, except for the fact that you browse the directory where the portlet WAR file was exported and copied to.

We followed the two-step process for deploying our Hello World portlet, placed it on a page on the portal and accessed the page with the result as seen in Figure 6-33.

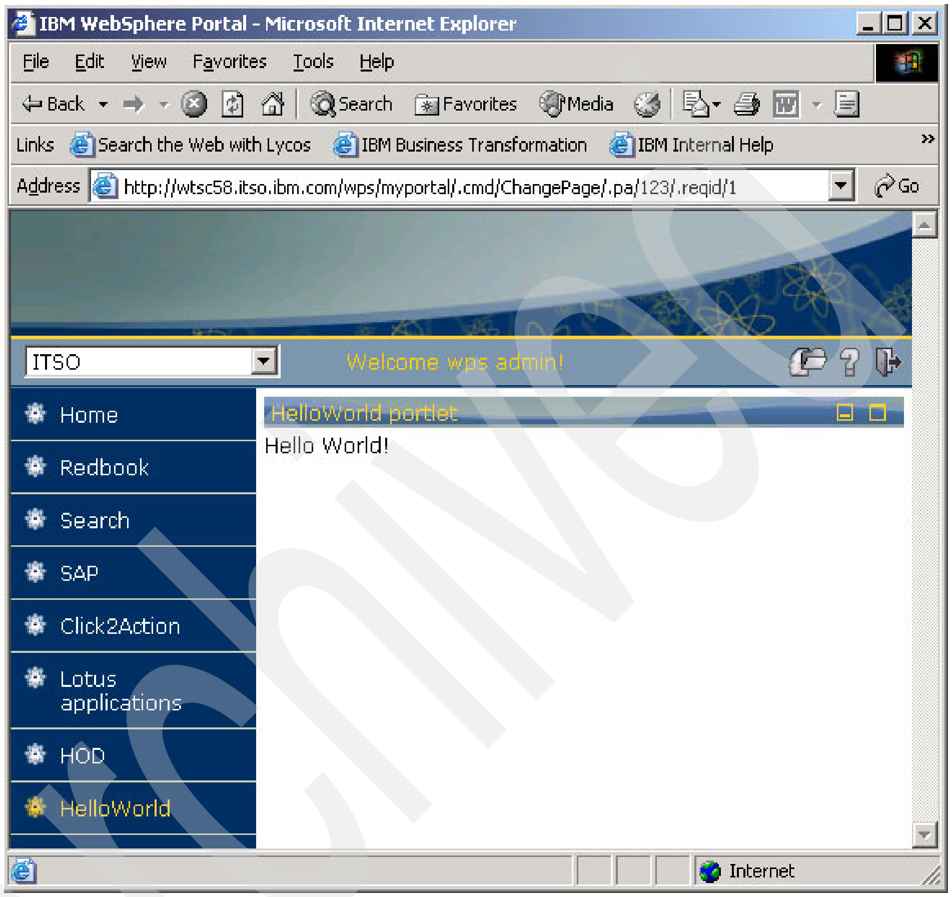

IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ChangePage/.pa/123/.reqid/1 Go
ITSO
Welcome wps admin!
Home
Redbook
Search
SAP
Click2Action
Lotus applications
HOD
HelloWorld
HelloWorld portlet
Hello World!
Internet

Figure 6-33 Our Hello World portlet deployed in the portal

6.4.5 Accessing CICS from a portlet

In this section we take a J2EE application called *Trader* that accesses an application running on CICS. The solution was developed for redpaper *From code to deployment: Connecting to CICS from WebSphere V4.01 for z/OS,* REDP0206, and run it under WebSphere Portal Server using the following two methods:

1. Use the iFrame portlet to access the Trader application from its browser based GUI.

2. Convert the Web application part of the J2EE application into a portlet.

Another option would be to use the `ServletInvokerPortlet`, that is installed but not deployed, to access and display the Web application from a portlet. The ServletInvokerPortlet can be configured with a URL to point to the Web page of a Web application. We did explore this option but at the time of writing we were not successful in getting this portlet to render correctly on z/OS.

The Trader application is available for download from:

`http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0206.html?Open`

It comes as packaged in a single file named **`Connector_code.zip.`** We downloaded the file and unzipped it to a directory and examined its contents. The unpacked file contains several directories: Completed Trader, Trader-Cobol, VisualAge® for Java and WebSphere Studio. Redpaper *From code to deployment: Connecting to CICS from WebSphere V4.01 for z/OS,* REDP0206, explains that the directory contents are as follows:

- Completed Trader: the J2EE application EAR file
- Trader-Cobol: code and JCL for the CICS application
- VisualAge for Java: commarea data for the J2EE EJB
- WebSphere Studio: WAR file and classes for the Web application

Figure 6-34 shows an overview of the complete Trader application.

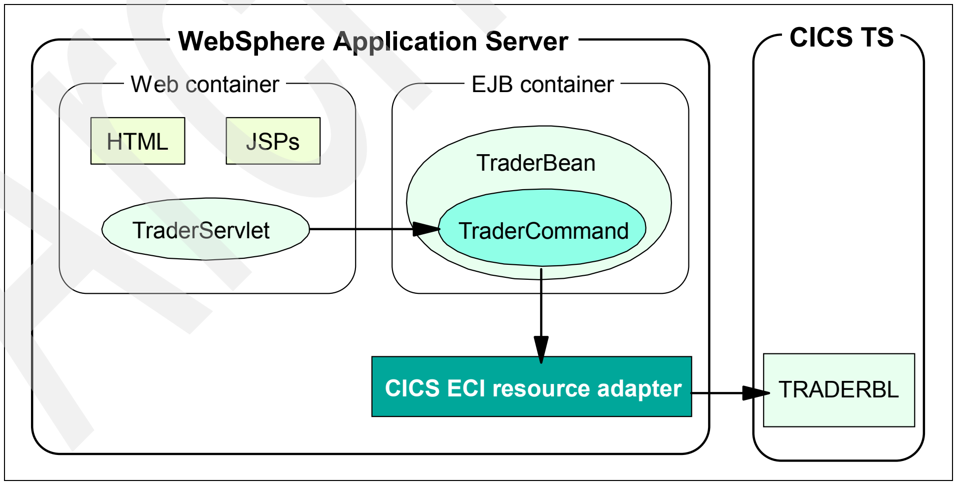
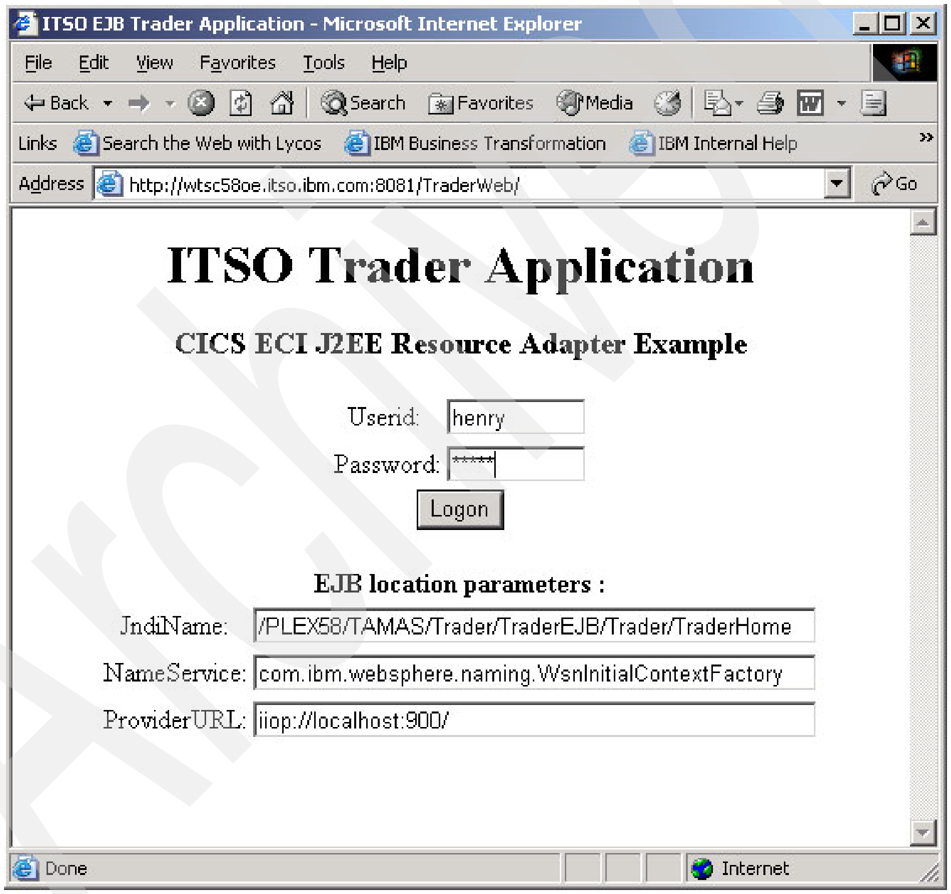


Figure 6-34 Trader application

Deployment of the complete application required the following prerequisites which we will not cover further in this book:

1. WebSphere Application Server V4
2. CTG V4.0.2 or CTG V5
3. CICS TS V1.3

Once installed and deployed in a separate J2EE server region on our WebSphere Application Server and CICS the GUI for the Trader application looks like the example shown in Figure 6-35.



ITSO EJB Trader Application - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address
Go
Done
Internet

http://wtsc58oe.itso.ibm.com:8081/TraderWeb/

ITSO Trader Application

CICS ECI J2EE Resource Adapter Example

Userid: henry

Password: *****

Logon

EJB location parameters :

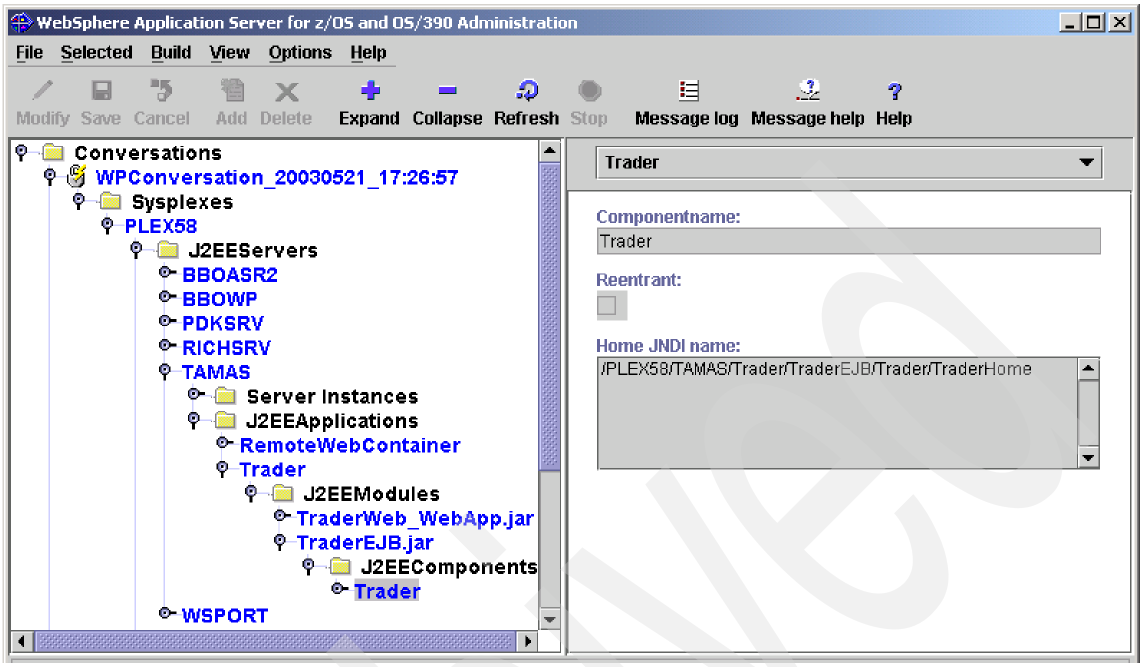
JndiName: /PLEX58/TAMAS/Trader/TraderEJB/Trader/TraderHome

NameService: com.ibm.websphere.naming.WsnInitialContextFactory

ProviderURL: iiop://localhost:900/

Figure 6-35 J2EE and CIC's Trader application GUI

We obtained the JNDI name parameter for locating the EJB from the systems management SMEUI GUI as shown in Figure 6-36 on page 226.


WebSphere Application Server for z/OS and OS/390 Administration
File Selected Build View Options Help
Modify Save Cancel Add Delete Expand Collapse Refresh Stop Message log Message help Help
Conversations
WPConversation_20030521_17:26:57
Sysplexes
PLEX58
J2EEServers
BBOASR2
BBOWP
PDKSRV
RICHSRV
TAMAS
Server Instances
J2EEApplications
RemoteWebContainer
Trader
J2EEModules
TraderWeb_WebApp.jar
TraderEJB.jar
J2EEComponents
Trader
WSPORT
Trader

Componentname: Trader

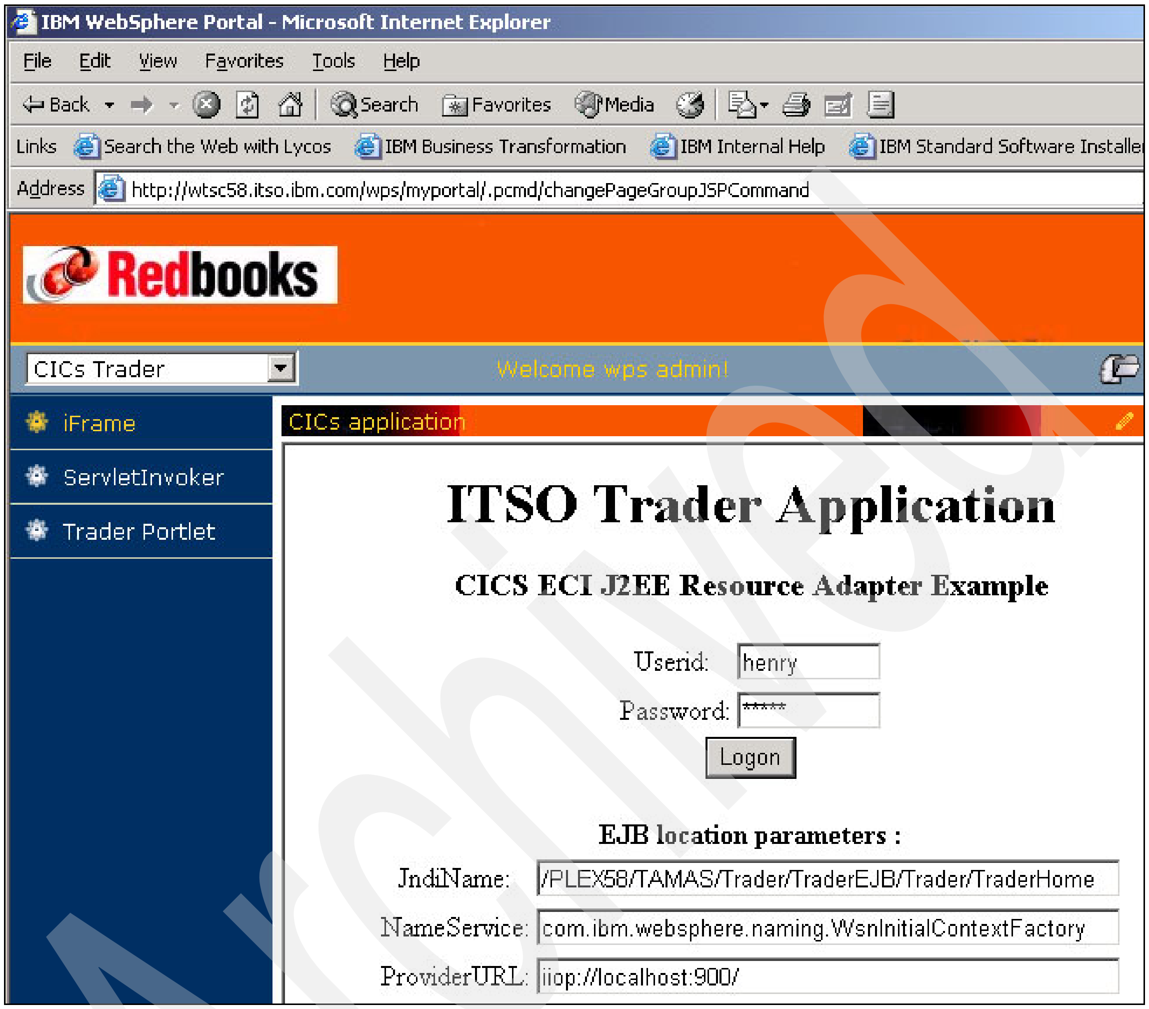
Reentrant: ☐

Home JNDI name: /PLEX58/TAMAS/Trader/TraderEJB/Trader/TraderHome

Figure 6-36 SMEUI view of TraderEJB home JNDI name

Using an iFrame to access Trader application from Portal

The iFrame portlet is one of the portlets that are installed and deployed during installation of the portal on z/OS. As its name implies it renders a target Web page by showing it inside an iFrame. The iFrame, which stands for *inline frame*, is still a fairly recent introduction to HTML and may not yet be supported by all browsers. We configured the iFrame portlet to point to URL `http://wtsc58oe.itso.ibm.com:8081/TraderWeb/` and used the logon screen shown in Figure 6-37 on page 227 to logon to the Trader application.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installer
Address

http://wtsc58.itso.ibm.com/wps/myportal/.pcmd/changePageGroupJSPCommand

Redbooks
CICs Trader
Welcome wps admin!
iFrame
ServletInvoker
Trader Portlet
CICs application

ITSO Trader Application

CICS ECI J2EE Resource Adapter Example

Userid:	henry
Password:	*****

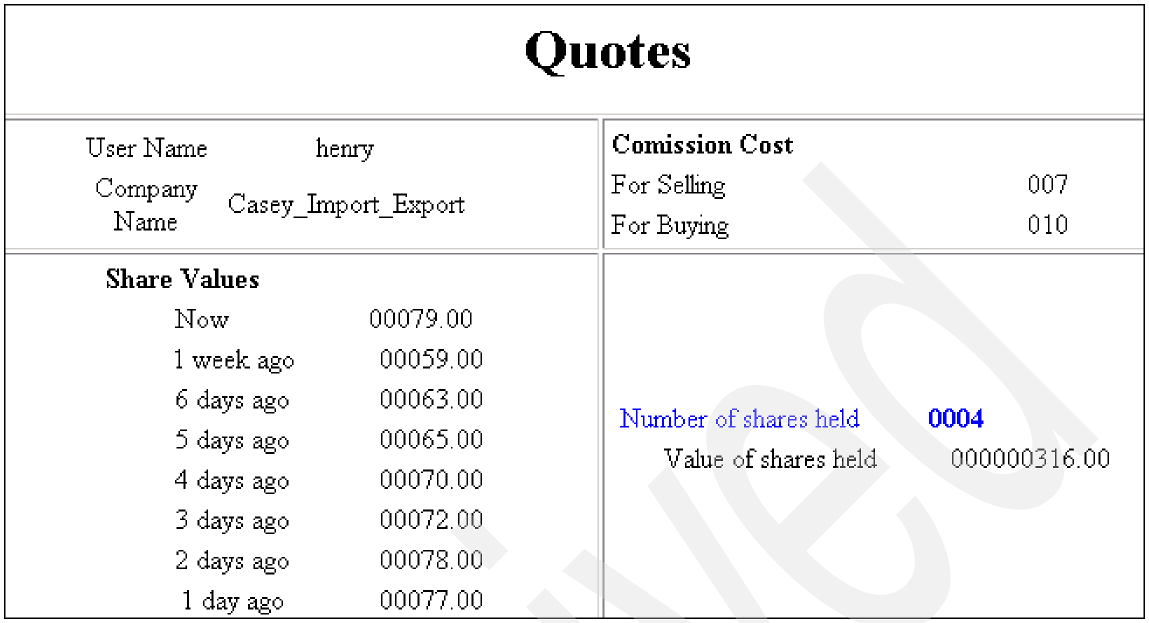
Logon

EJB location parameters :

JndiName:	/PLEX58/TAMAS/Trader/TraderEJB/Trader/TraderHome
NameService:	com.ibm.websphere.naming.WsnInitialContextFactory
ProviderURL:	iiop://localhost:900/

Figure 6-37 Logging onto Trader using the iFrame portlet

Since the Trader Web application uses quite straightforward HTML to render its simple user interface pages, the iFrame portlet handles the application quite well. Figure 6-38 on page 228 shows another example screen from one of the application's pages.


Quotes

User Name	henry	**Comission Cost**	
Company Name	Casey_Import_Export	For Selling	007
		For Buying	010

Share Values

Now	00079.00
1 week ago	00059.00
6 days ago	00063.00
5 days ago	00065.00
4 days ago	00070.00
3 days ago	00072.00
2 days ago	00078.00
1 day ago	00077.00

Number of shares held	**0004**
Value of shares held	000000316.00

Figure 6-38 Using the Trader application with the iFrame portlet

Converting Trader Web application to a portlet

As we have seen in Figure 6-34 on page 224, the Trader J2EE application consists of an EJB part and a Web application part. To transform this into a portlet we decided to leave untouched the EJB and concentrate on converting the Web application, consisting of servlets and JSPs, into a portlet. Some would call this *portalizing*.

For this we left the Web application and EJB installation untouched, running on another J2EE server on the same application server as portal. Figure 6-39 on page 229 shows an example of what we wanted to achieve by portalizing the Web application into a portlet called TraderPortlet, in comparison to using the iFrame portlet.

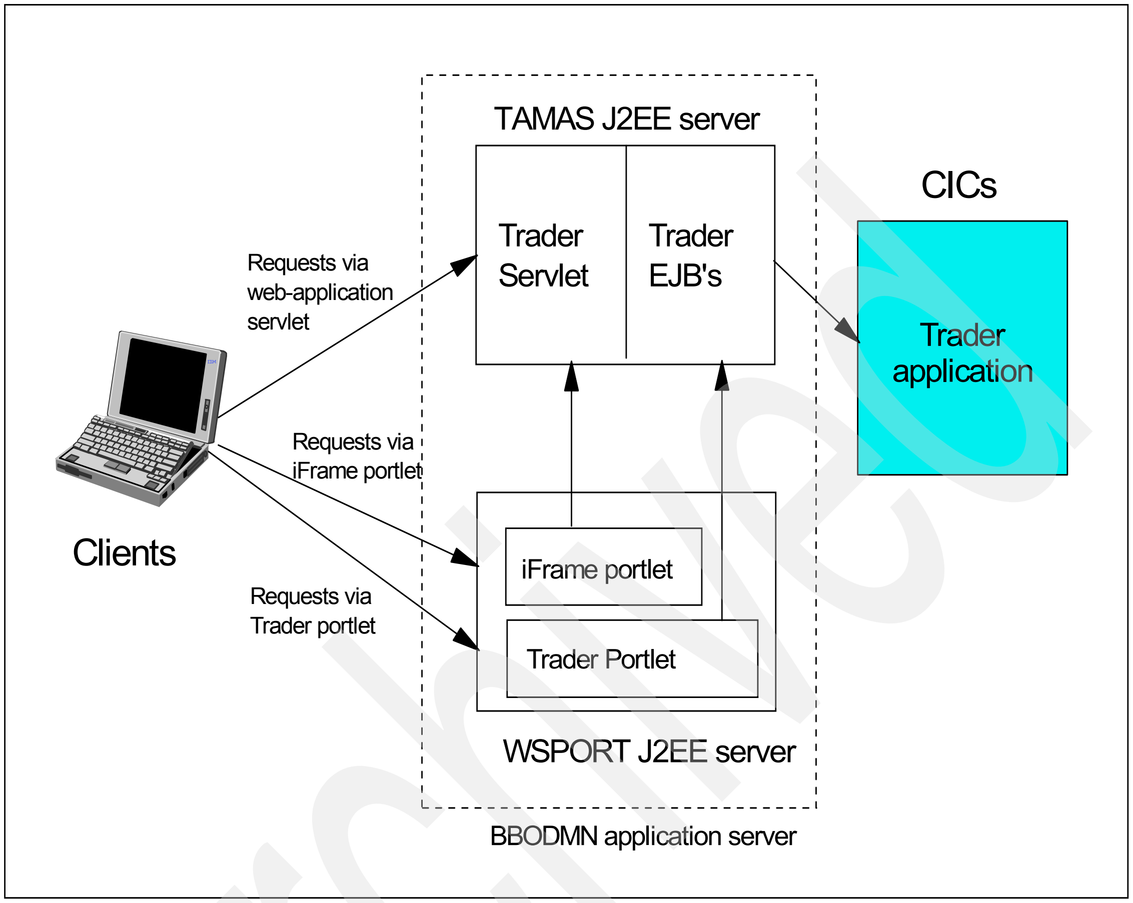
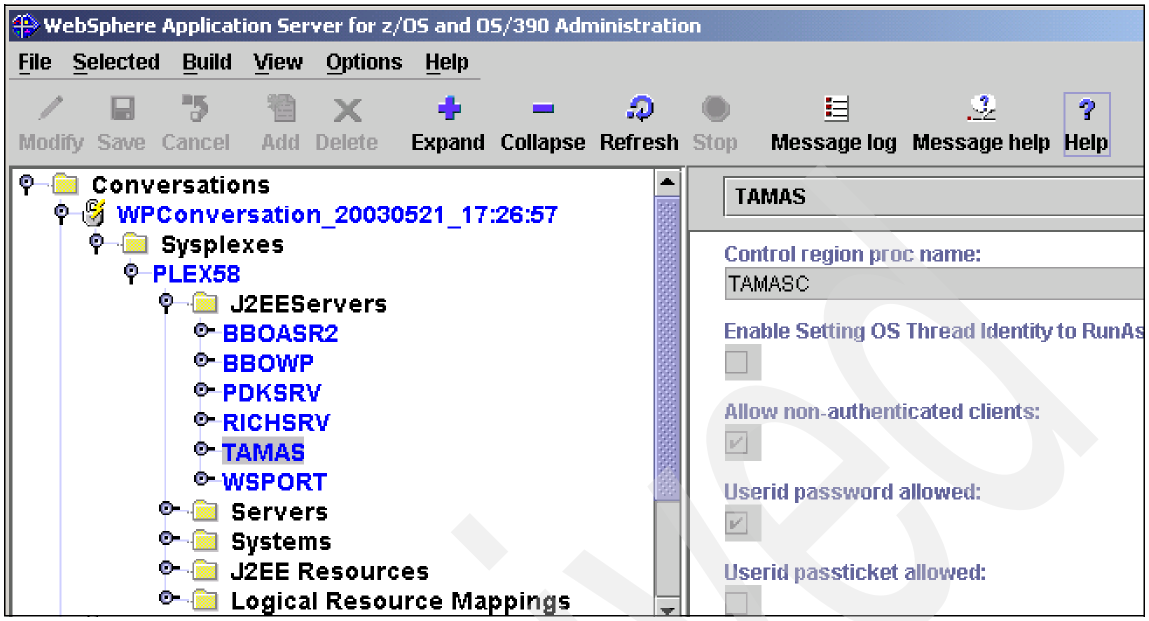


Figure 6-39 TraderPortlet and iFrame portlet compared

A view from the systems management GUI SMEUI of our J2EE servers TAMAS and WSPORT is shown in Figure 6-40 on page 230. Note that for this project we did not enable any J2EE EJBROLES based security, so that by default our portal server ID was running with read access to EJBROLE. For a further discussion on this refer to Chapter 7, “Custom User Registry” on page 285.


WebSphere Application Server for z/OS and OS/390 Administration
File Selected Build View Options Help
Modify Save Cancel Add Delete Expand Collapse Refresh Stop Message log Message help Help

Conversations
WPConversation_20030521_17:26:57
Sysplexes
PLEX58
J2EEServers
BBOASR2
BBOWP
PDKSRV
RICHSRV
TAMAS
WSPORT
Servers
Systems
J2EE Resources
Logical Resource Mappings

TAMAS

Control region proc name: TAMASC

Enable Setting OS Thread Identity to RunAs: ☐

Allow non-authenticated clients: ✓

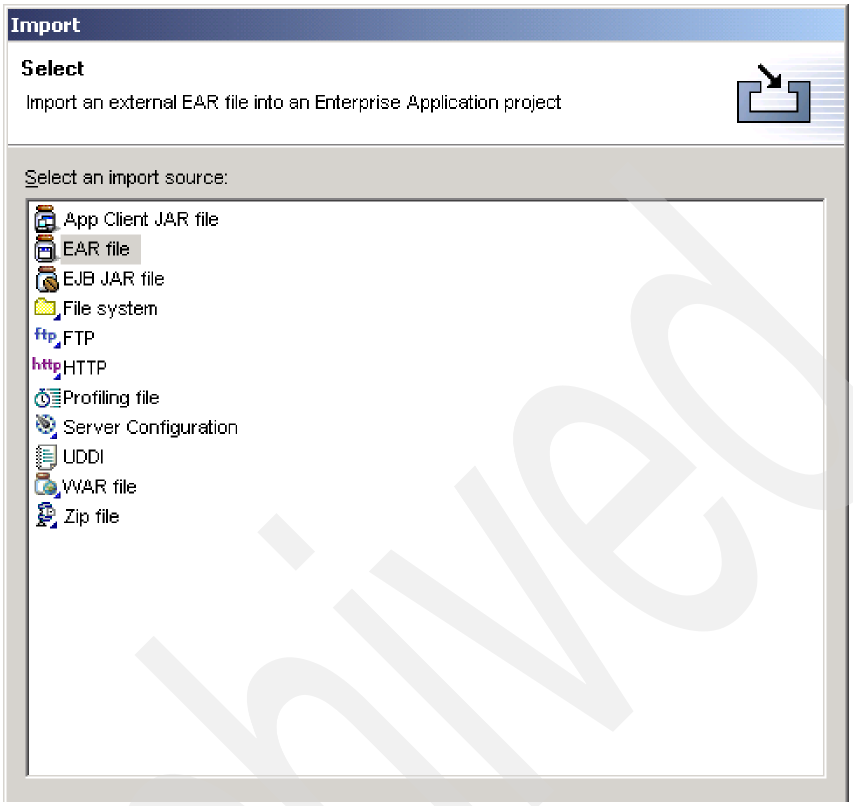
Userid password allowed: ✓

Userid passticket allowed: ☐

Figure 6-40 SMEUI view of J2EE servers

Our example does not represent a complete set of instructions on how to portalize a Web application, but just shows the steps we performed to get the example working with WebSphere Portal.

We started by importing the `Completed_Trader.ear` file into Application Developer using the import wizard as seen in Figure 6-41 on page 231.


Import

Select

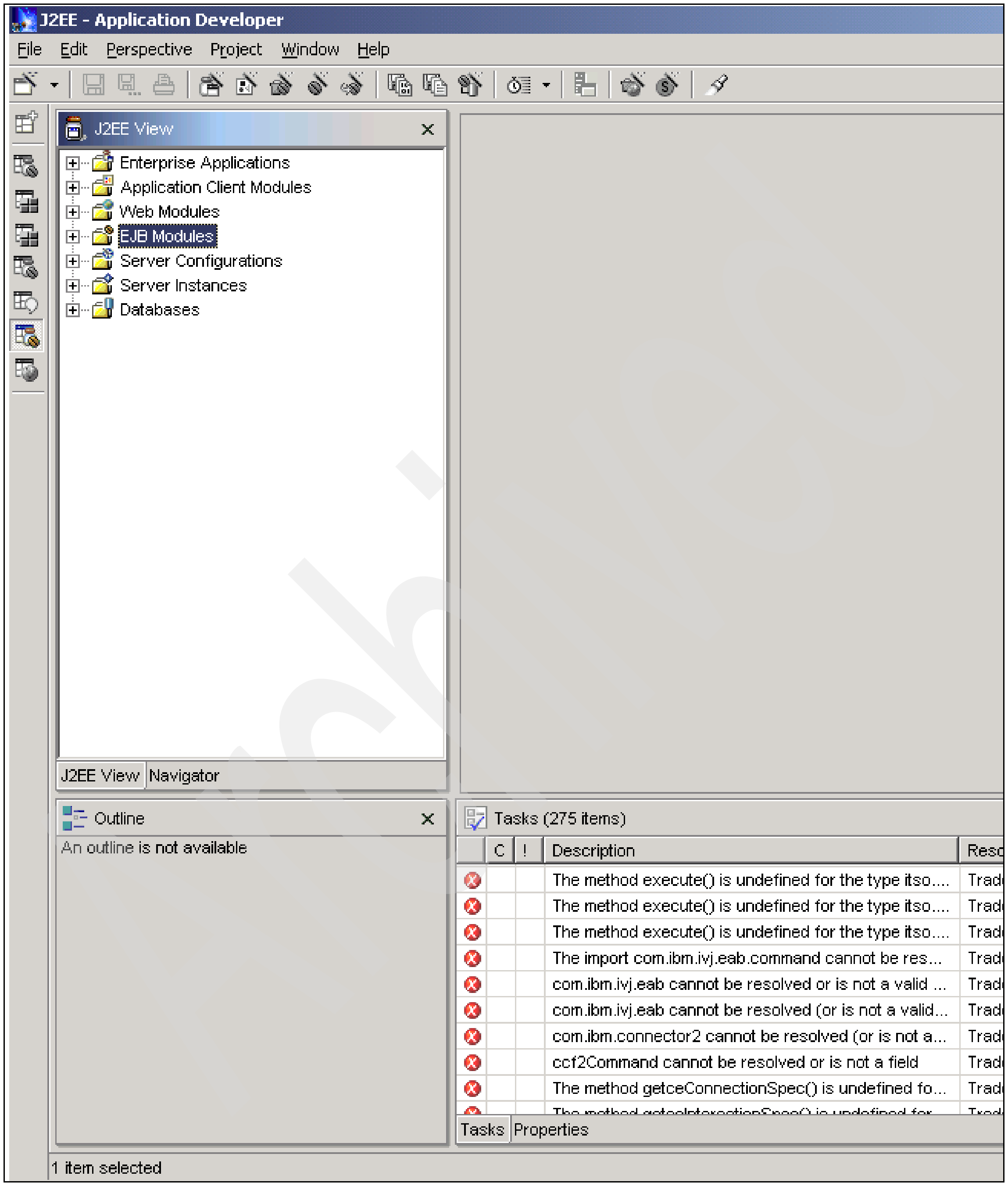
Import an external EAR file into an Enterprise Application project

Select an import source:

- App Client JAR file
- EAR file
- EJB JAR file
- File system
- FTP
- HTTP
- Profiling file
- Server Configuration
- UDDI
- WAR file
- Zip file

Figure 6-41 Import J2EE EAR wizard.

After completing this, then as shown from the J2EE perspective in Figure 6-42 on page 232 we were left with 275 project problems which Application Developer calls tasks.



Tasks (275 items)

C	!	Description	Reso
		The method execute() is undefined for the type itso....	Trad
		The method execute() is undefined for the type itso....	Trad
		The method execute() is undefined for the type itso....	Trad
		The import com.ibm.ivj.eab.command cannot be res...	Trad
		com.ibm.ivj.eab cannot be resolved or is not a valid ...	Trad
		com.ibm.ivj.eab cannot be resolved (or is not a valid...	Trad
		com.ibm.connector2 cannot be resolved (or is not a...	Trad
		ccf2Command cannot be resolved or is not a field	Trad
		The method getceConnectionSpec() is undefined fo...	Trad

Figure 6-42 Tasks outstanding after importing Completed_Trader.ear

Not to worry, at this stage there were undoubtedly missing JAR files and since we would work with only half of the project, the Web application WAR, we decided to filter them out by selecting **`Filter Tasks`**->**`On selected resource`** only as seen in Figure 6-43.

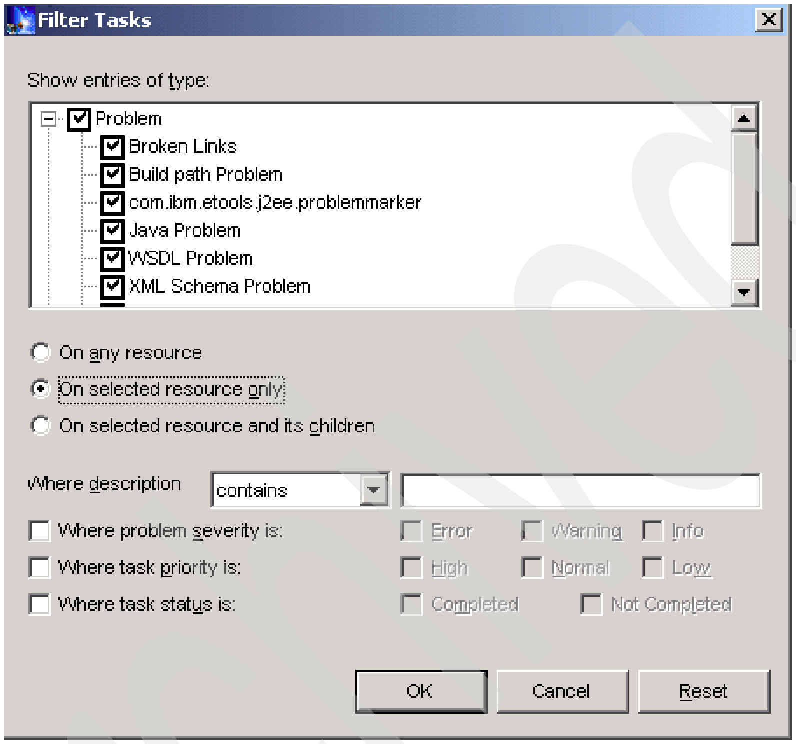


Figure 6-43 Filter tasks option

Our next step was to create the Portlet Application project in Application Developer by switching to a Portlet perspective, then right-clicking in the Navigator frame and selecting **`New`** -> **`Project`** -> **`Portlet application project`** as seen in Figure 6-44 on page 234.

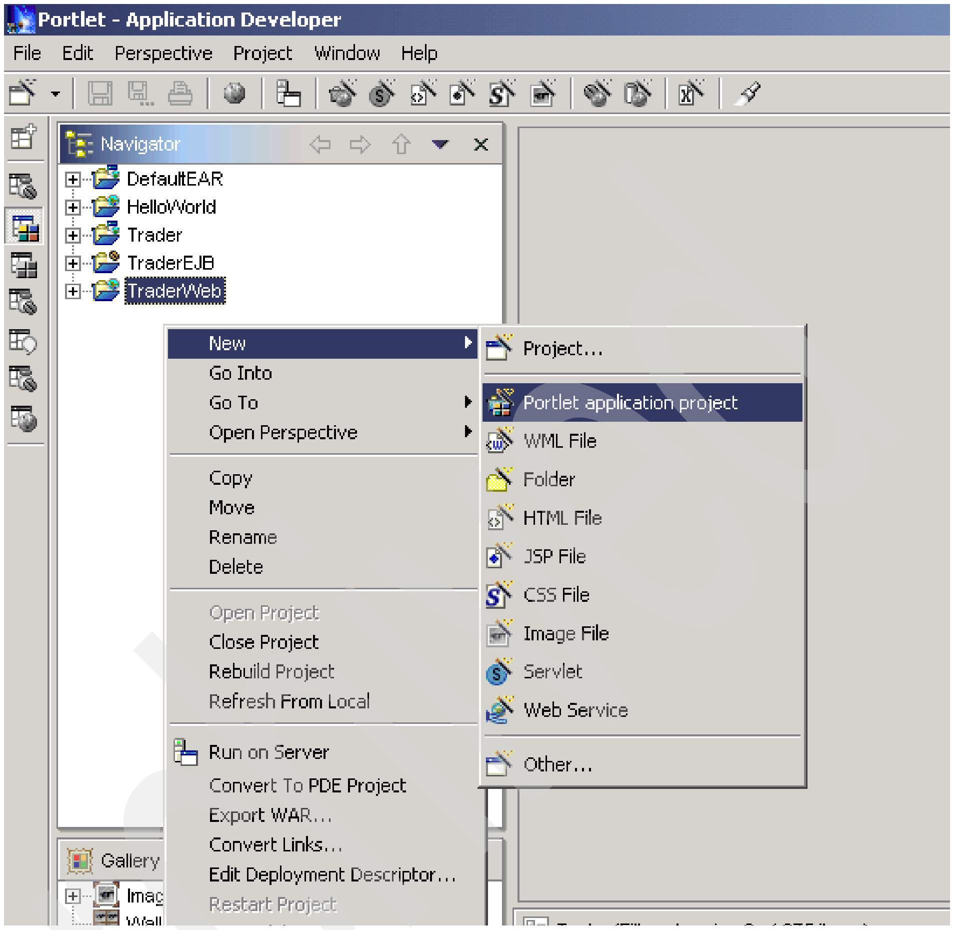
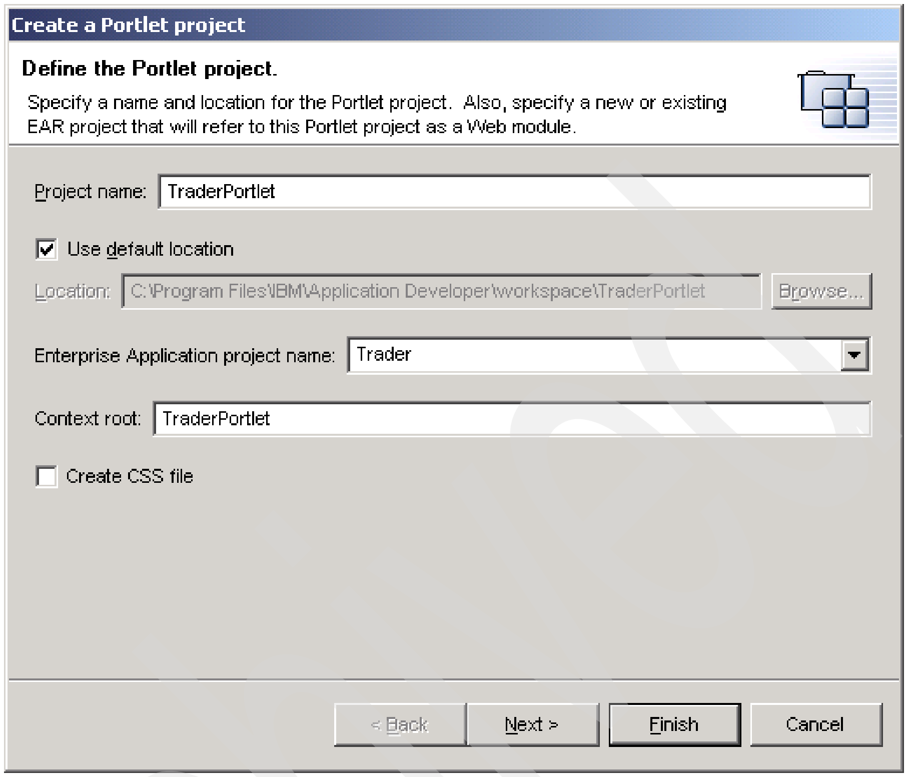


Figure 6-44 Creating a Portlet application project

Figure 6-45 on page 235 shows the settings we provided to define our Portlet project which we decided to call `TraderPortlet`. After entering the project's settings we clicked on **`Finish`** since we did not want Application Developer to add portlets for us, because we will use the `TraderWeb.war` Web application as our base for a portlet.


Create a Portlet project

Define the Portlet project.

Specify a name and location for the Portlet project. Also, specify a new or existing EAR project that will refer to this Portlet project as a Web module.

Project name: TraderPortlet

[✓] Use default location

Location: C:\Program Files\IBM\Application Developer\workspace\TraderPortlet

Enterprise Application project name: Trader

Context root: TraderPortlet

[ ] Create CSS file

Figure 6-45 Define the Portlet project

After clicking on **Finish**, Application Developer creates and populates the project for us as shown in Figure 6-46 on page 236.

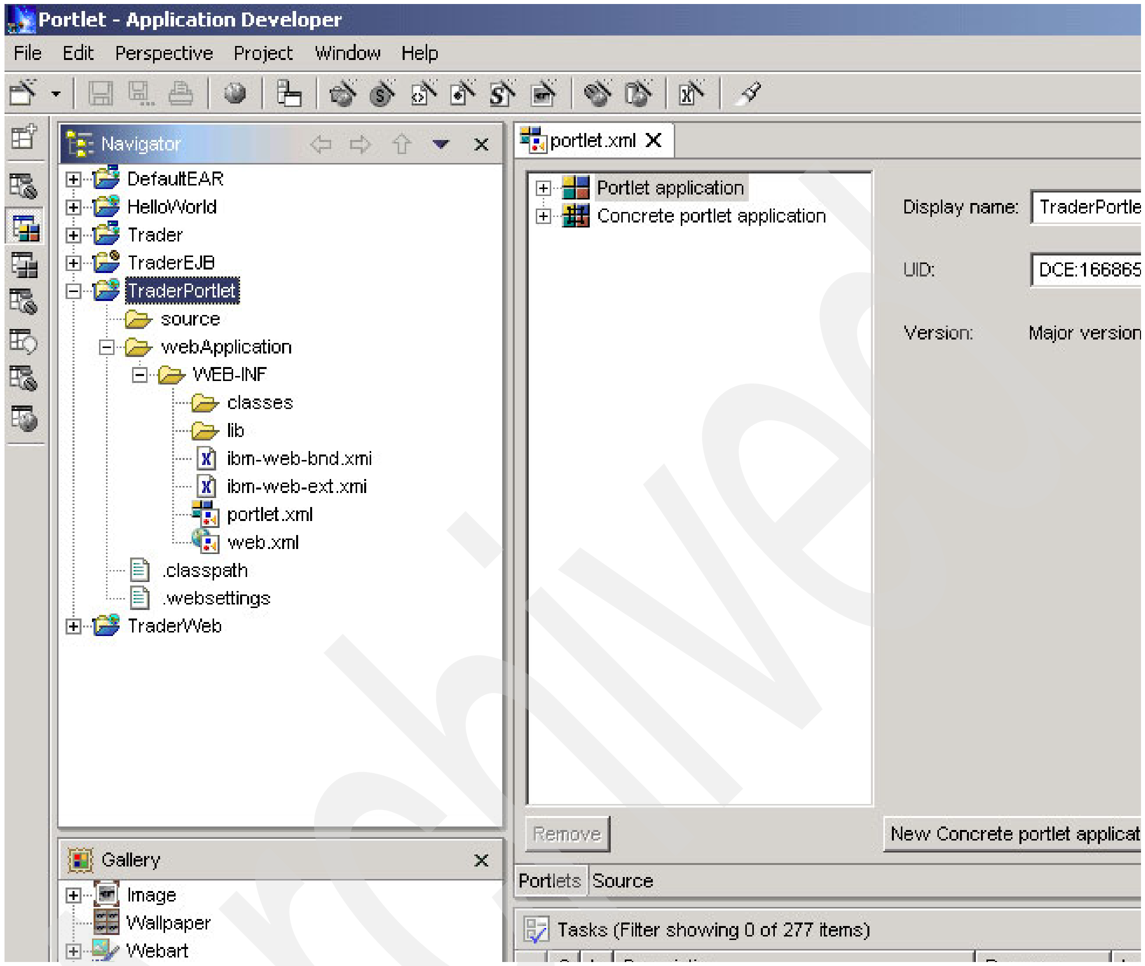


Figure 6-46 TraderPortlet project

Next we added TraderWeb.war to the project by clicking on the **TraderPortlet** project name and selecting **File** -> **Import** -> **WAR** file, as shown in Figure 6-47 on page 237.

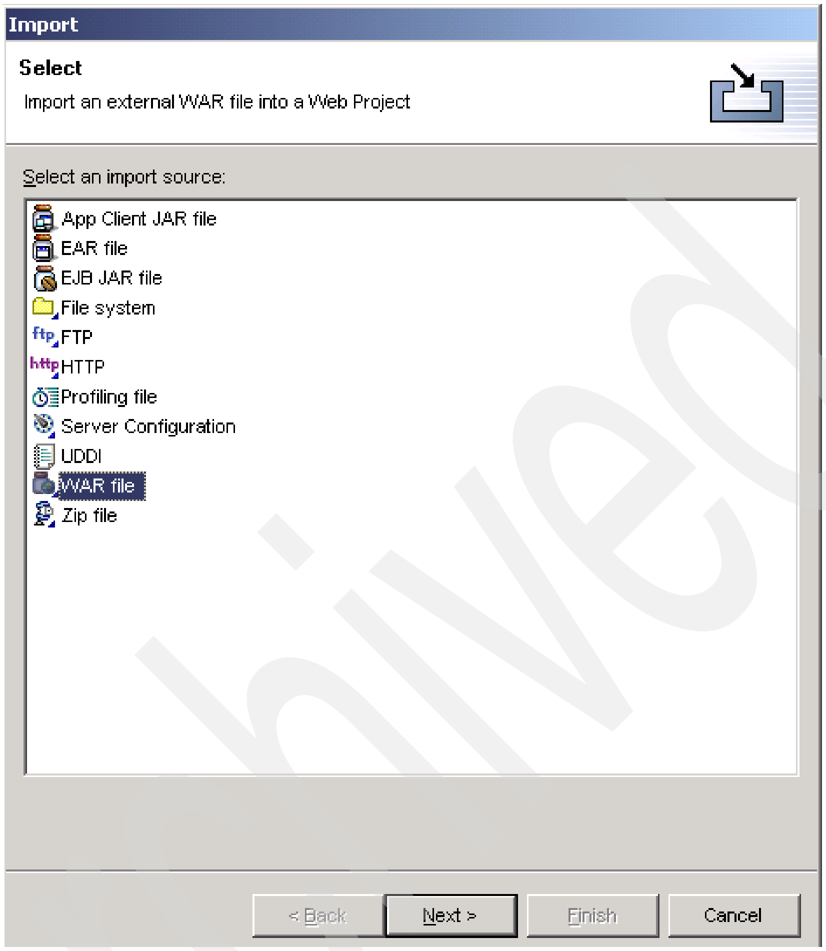
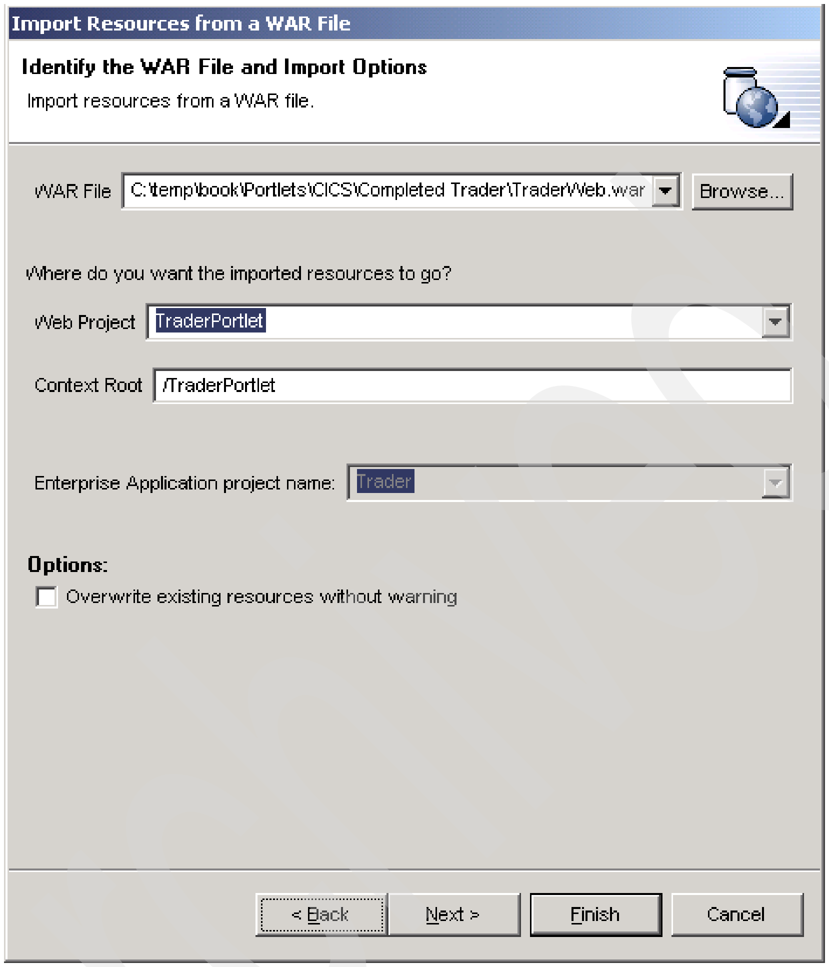


Figure 6-47 Import WAR

Figure 6-48 on page 238 shows the location of the WAR file and the options we set for the import.


Import Resources from a WAR File

Identify the WAR File and Import Options

Import resources from a WAR file.

WAR File: C:\temp\book\Portlets\CICS\Completed Trader\TraderWeb.war

Where do you want the imported resources to go?

Web Project: TraderPortlet

Context Root: /TraderPortlet

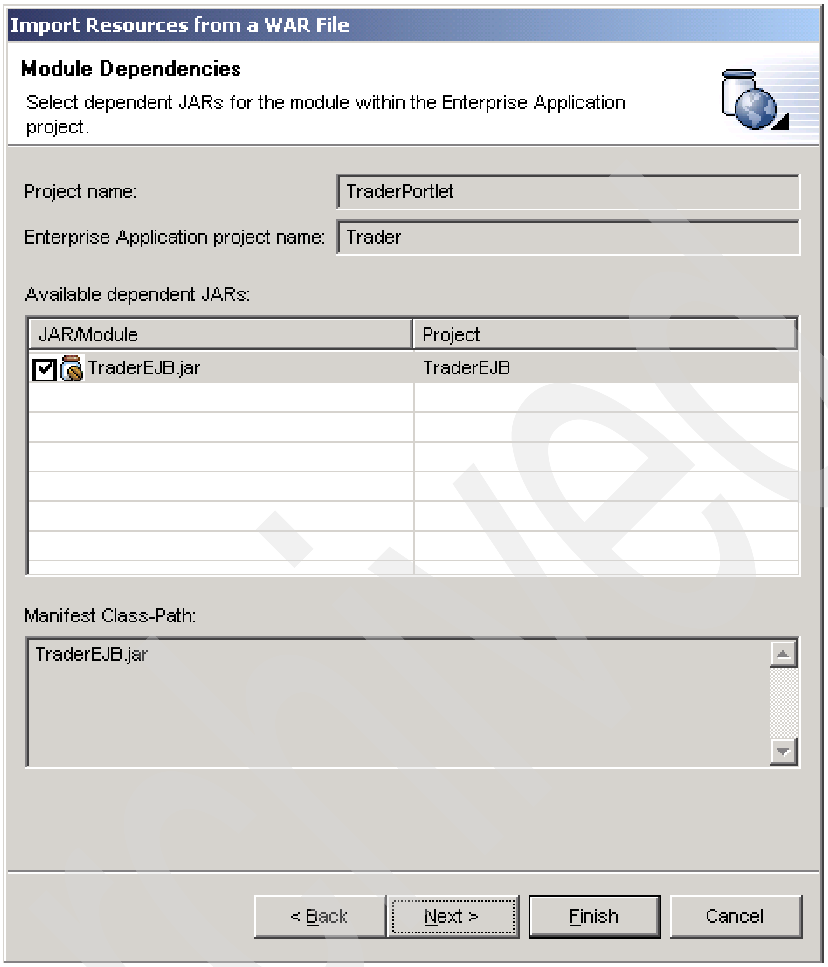
Enterprise Application project name: Trader

Options:

- [ ] Overwrite existing resources without warning

Figure 6-48 Import options for the WAR file

Finally we set the Module dependencies, which will update Application Developer's build and runtime classpaths, by clicking the checkbox for **`TraderEJB.jar`** as seen in Figure 6-49 on page 239.



Module Dependencies

Select dependent JARs for the module within the Enterprise Application project.

Project name: TraderPortlet

Enterprise Application project name: Trader

Available dependent JARs:

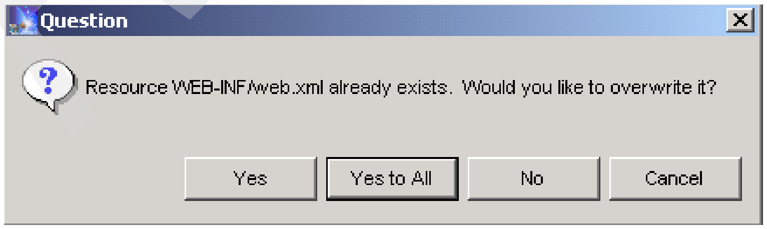
JAR/Module	Project
✓ TraderEJB.jar	TraderEJB

Manifest Class-Path:

TraderEJB.jar

Figure 6-49 Set the module dependencies

We then clicked on **`Finish`** and also clicked on **`Yes`** to the dialog window shown in Figure 6-50.



Resource WEB-INF/web.xml already exists. Would you like to overwrite it?

Figure 6-50 Resource dialog

Application Developer completes the import of the TraderWeb Web application WAR file into our portlet project as shown in Figure 6-51.

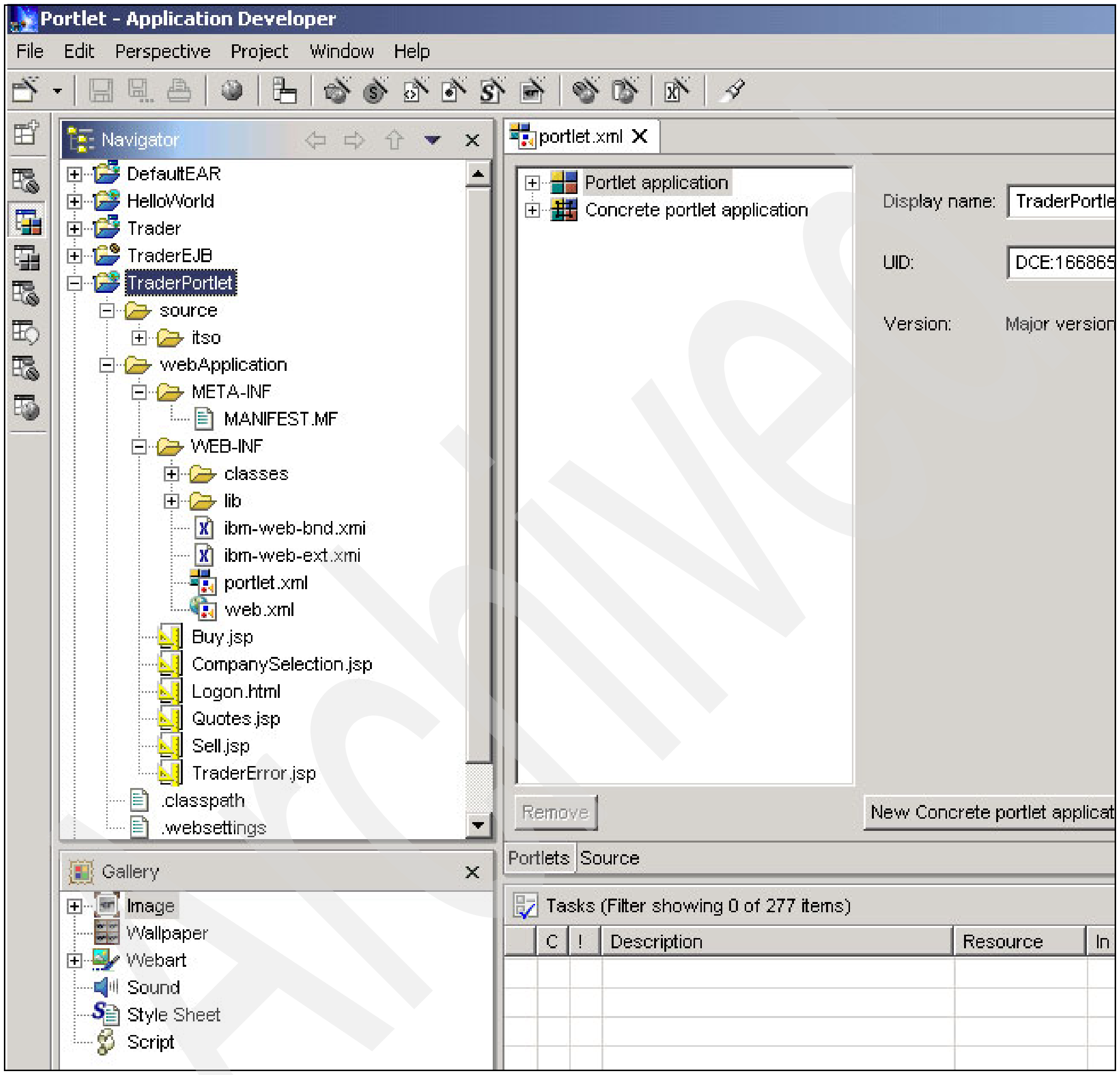


Figure 6-51 Portlet application project with imported code from TraderWeb.war

Modifying the imported servlet to create a portlet

We now need to make changes to the imported servlet files to convert them into a portlet. The main items we needed to consider, examine and possibly change are the following.

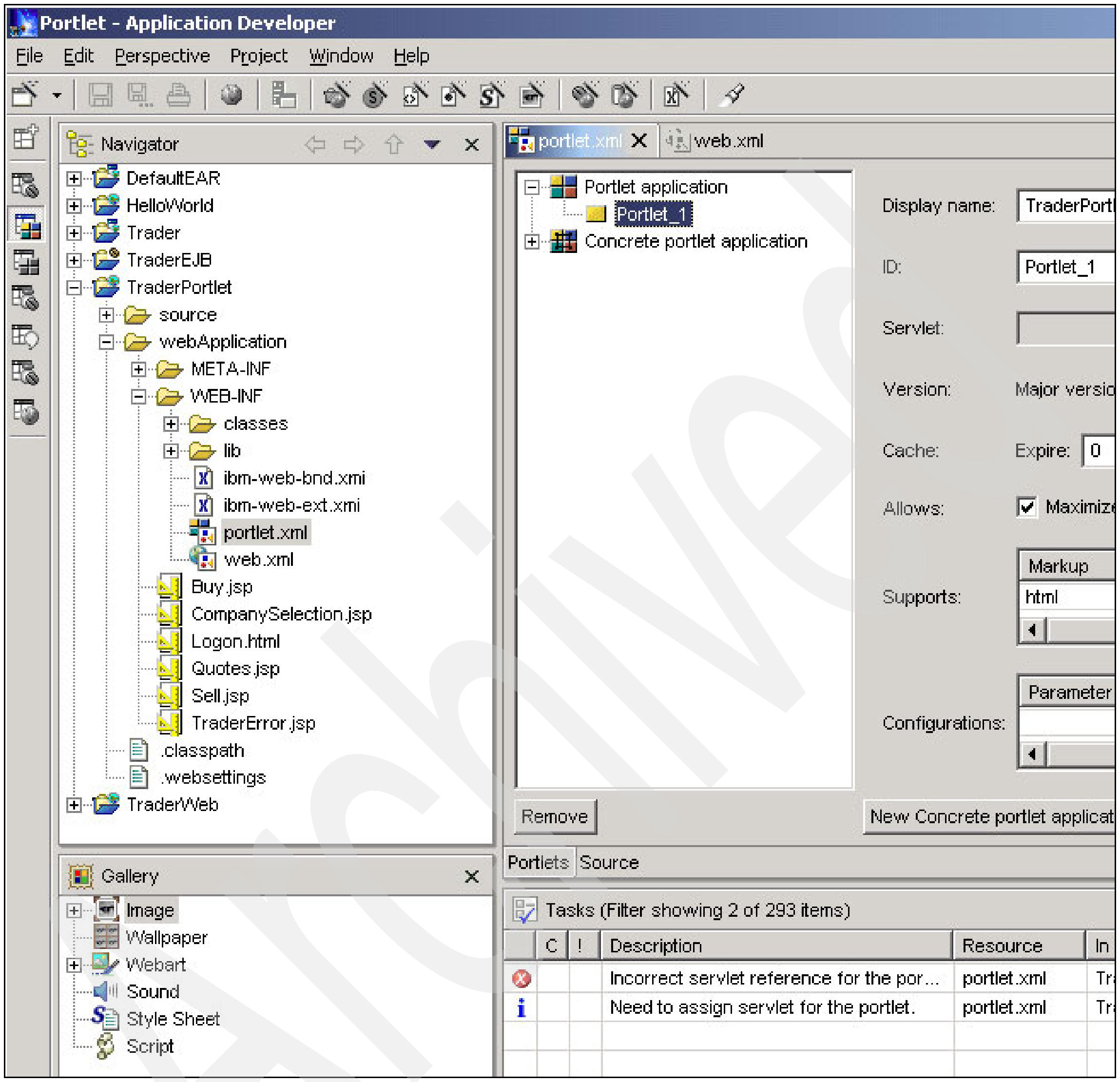
1. portlet.xml: Assign a servlet to be used for the portlet.
2. HTML pages: Convert to JSPs.
3. JSP source: Ensure only page fragments are in the JSP, use the Portal tag library as desired, parameterize the ACTION field of any FORMS, and ensure the INPUT field of any forms is unique in the portal namespace.
4. Java source: Use Portlet API instead of Servlet API.
5. web.xml: Update references.
6. Miscellaneous considerations for changing a servlet to a portlet.

Note that as we have already said, this list is not meant to represent all the items that need to be considered when portalizing a Web application, but it was enough for our project, and we explain each of the things we did in more detail in the following steps.

1) Changing Portlet.xml

Servlets provide descriptor file `web.xml` in directory WEB-INF. In addition to this file portlets need to provide their own descriptor file `portlet.xml`. Parameters in portlet.xml typically include the servlet ID, concrete portlet ID, mark-up languages supported, title, and configuration options.

Using Application Developer we clicked on file **`portlet.xml`** in the WEB-INF directory, and noted that there were two tasks listed, complaining about the lack of a servlet reference. We expanded the Portlet application in the right hand Portlets view frame, and selected Portlet_1 as shown in Figure 6-52 on page 242.


Portlet - Application Developer
File Edit Perspective Project Window Help
Navigator
Gallery
portlet.xml
web.xml
Remove
New Concrete portlet applicat
Portlets Source

```
DefaultEAR
HelloWorld
Trader
TraderEJB
TraderPortlet
    source
    webApplication
        META-INF
        WEB-INF
            classes
            lib
            ibm-web-bnd.xmi
            ibm-web-ext.xmi
            portlet.xml
            web.xml
        Buy.jsp
        CompanySelection.jsp
        Logon.html
        Quotes.jsp
        Sell.jsp
        TraderError.jsp
    .classpath
    .websettings
TraderWeb
```

```
Image
Wallpaper
Webart
Sound
Style Sheet
Script
```

```
Portlet application
    Portlet_1
Concrete portlet application
```

Display name:	TraderPortl
ID:	Portlet_1
Servlet:	
Version:	Major versio
Cache:	Expire: 0
Allows:	✓ Maximize
Supports:	Markup: html
Configurations:	Parameter

Tasks (Filter showing 2 of 293 items)

	C	!	Description	Resource	In
✖			Incorrect servlet reference for the por...	portlet.xml	Tr
i			Need to assign servlet for the portlet.	portlet.xml	Tr

Figure 6-52 Select Portlet_1 to assign a servlet to the portlet

Next we clicked on the **Browse** button next to the grayed out Servlet list box, and in the pop-up dialog window selected `TraderServlet` which was the only servlet listed, as shown in Figure 6-52 on page 242.

.

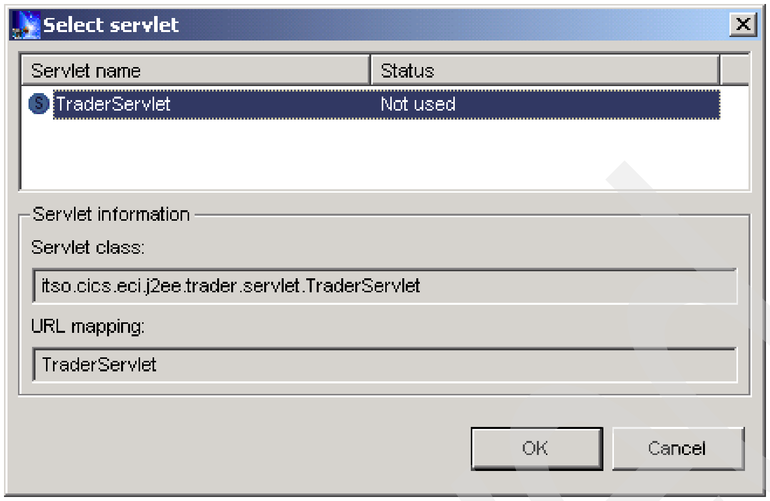
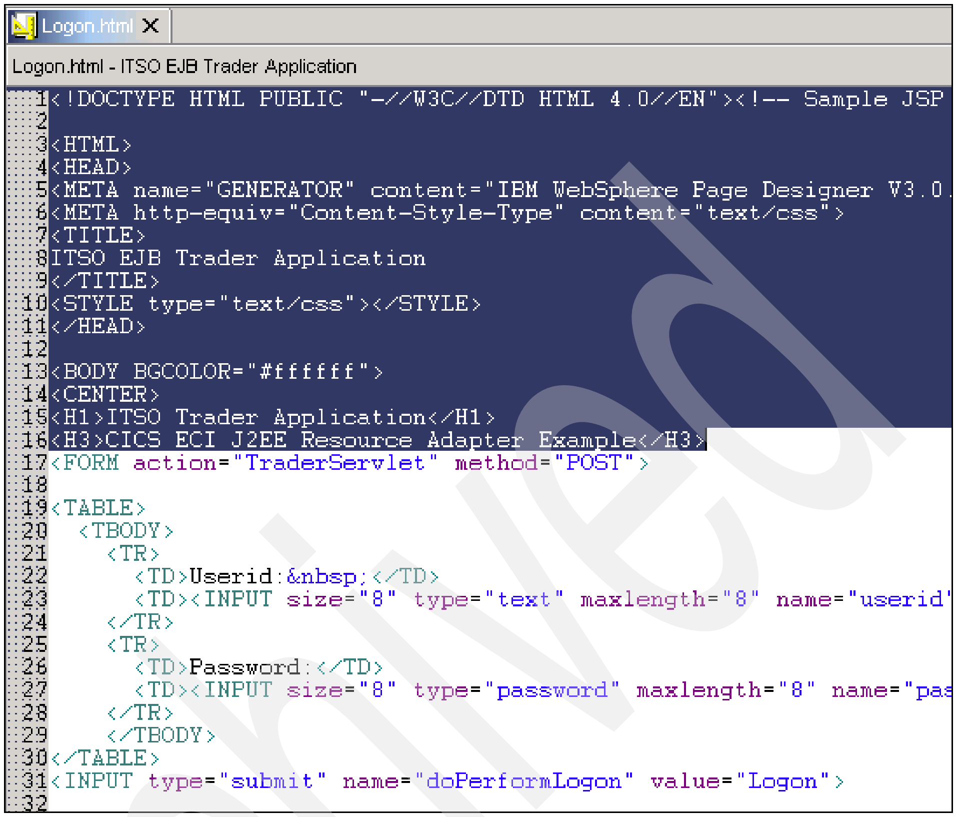


Figure 6-53 Select servlet to assign to the portlet

Click on **`OK`**, then **`File->Save portlet.xml`** and Application Developer updates the project and the Tasks listed for portlet.xml should have cleared.

2) Changing the HTML pages

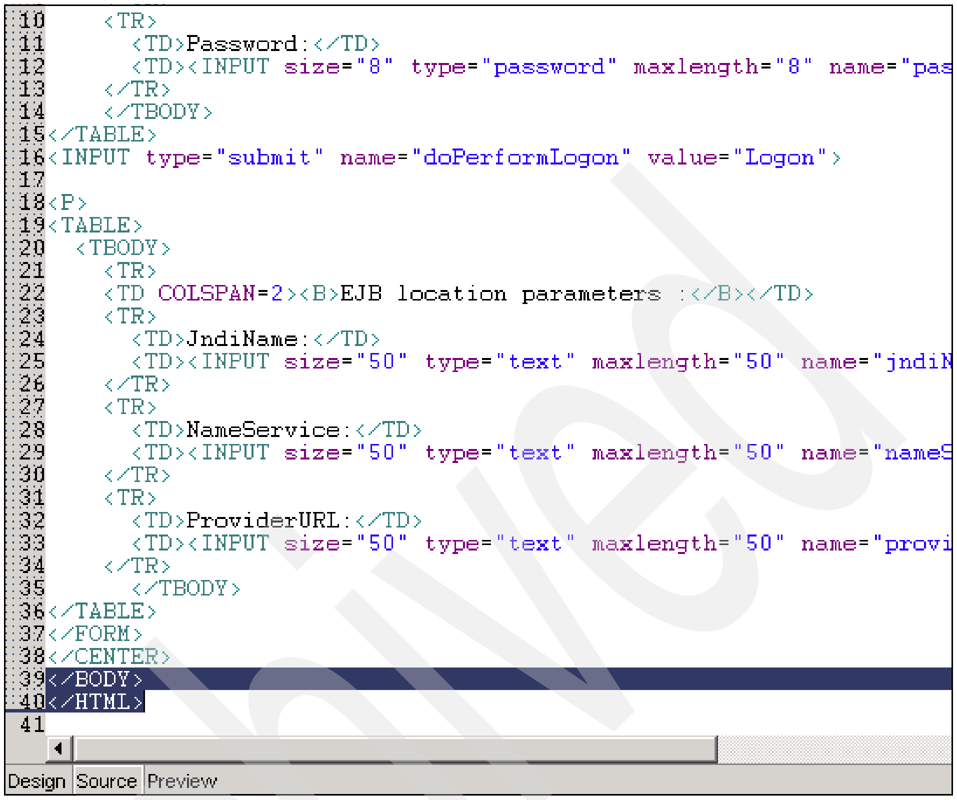
Because WebSphere Portal aggregates multiple portlet JSPs into a single Web page, any existing HTML pages need to be converted to JSPs containing only page fragments that are rendered from the Portlet Java code. The Trader Web application servlet had a single HTML page `Logon.html` that we converted, bit in general this would need to be done for all HTML pages making up a Web application. Figure 6-54 on page 244 shows the top page level tags in `Logon.html` ready for deletion.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"><!-- Sample JSP

<HTML>
<HEAD>
<META name="GENERATOR" content="IBM WebSphere Page Designer V3.0.
<META http-equiv="Content-Style-Type" content="text/css">
<TITLE>
ITSO EJB Trader Application
</TITLE>
<STYLE type="text/css"></STYLE>
</HEAD>

<BODY BGCOLOR="#ffffff">
<CENTER>
<H1>ITSO Trader Application</H1>
<H3>CICS ECI J2EE Resource Adapter Example</H3>
<FORM action="TraderServlet" method="POST">

<TABLE>
  <TBODY>
    <TR>
      <TD>Userid:&nbsp;</TD>
      <TD><INPUT size="8" type="text" maxlength="8" name="userid"
    </TR>
    <TR>
      <TD>Password:</TD>
      <TD><INPUT size="8" type="password" maxlength="8" name="pas
    </TR>
    </TBODY>
</TABLE>
<INPUT type="submit" name="doPerformLogon" value="Logon">

```

Figure 6-54 Deleting page level tags from the HTML

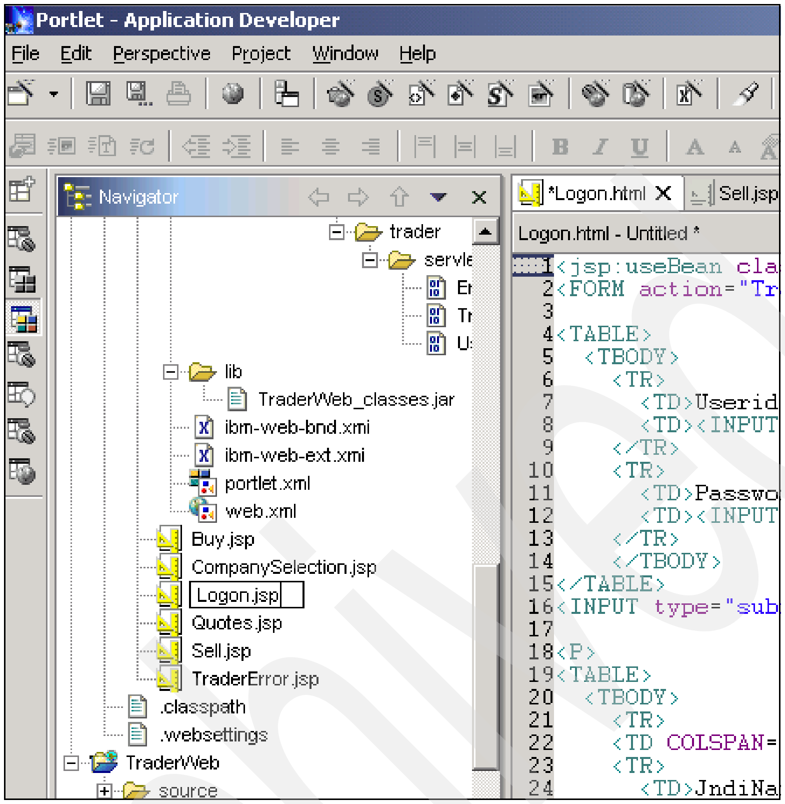
Figure 6-55 on page 245 shows the bottom page level tags in `Logon.html` ready for deletion.



```
    <TR>
      <TD>Password:</TD>
      <TD><INPUT size="8" type="password" maxlength="8" name="pas
    </TR>
    </TBODY>
</TABLE>
<INPUT type="submit" name="doPerformLogon" value="Logon">

<P>
<TABLE>
  <TBODY>
    <TR>
    <TD COLSPAN=2><B>EJB location parameters :</B></TD>
    <TR>
      <TD>JndiName:</TD>
      <TD><INPUT size="50" type="text" maxlength="50" name="jndiN
    </TR>
    <TR>
      <TD>NameService:</TD>
      <TD><INPUT size="50" type="text" maxlength="50" name="nameS
    </TR>
    <TR>
      <TD>ProviderURL:</TD>
      <TD><INPUT size="50" type="text" maxlength="50" name="provi
    </TR>
      </TBODY>
</TABLE>
</FORM>
</CENTER>
</BODY>
</HTML>
```

Figure 6-55 Deleting page level tags from the HTML

The next to last step was to rename the HTML file from `Logon.html` to `Logon.jsp` as seen in Figure 6-56 on page 246.


Portlet - Application Developer
File Edit Perspective Project Window Help
Navigator
*Logon.html
Sell.jsp
Logon.html - Untitled *

```
trader
  servle
    Er
    Tr
    U:
lib
  TraderWeb_classes.jar
ibm-web-bnd.xmi
ibm-web-ext.xmi
portlet.xml
web.xml
Buy.jsp
CompanySelection.jsp
Logon.jsp
Quotes.jsp
Sell.jsp
TraderError.jsp
.classpath
.websettings
TraderWeb
  source
```

```
<jsp:useBean cla
<FORM action="Tr

<TABLE>
  <TBODY>
    <TR>
      <TD>Userid
      <TD><INPUT
    </TR>
    <TR>
      <TD>Passwo
      <TD><INPUT
    </TR>
    </TBODY>
</TABLE>
<INPUT type="sub

<P>
<TABLE>
  <TBODY>
    <TR>
    <TD COLSPAN=
    <TR>
      <TD>JndiNa
```

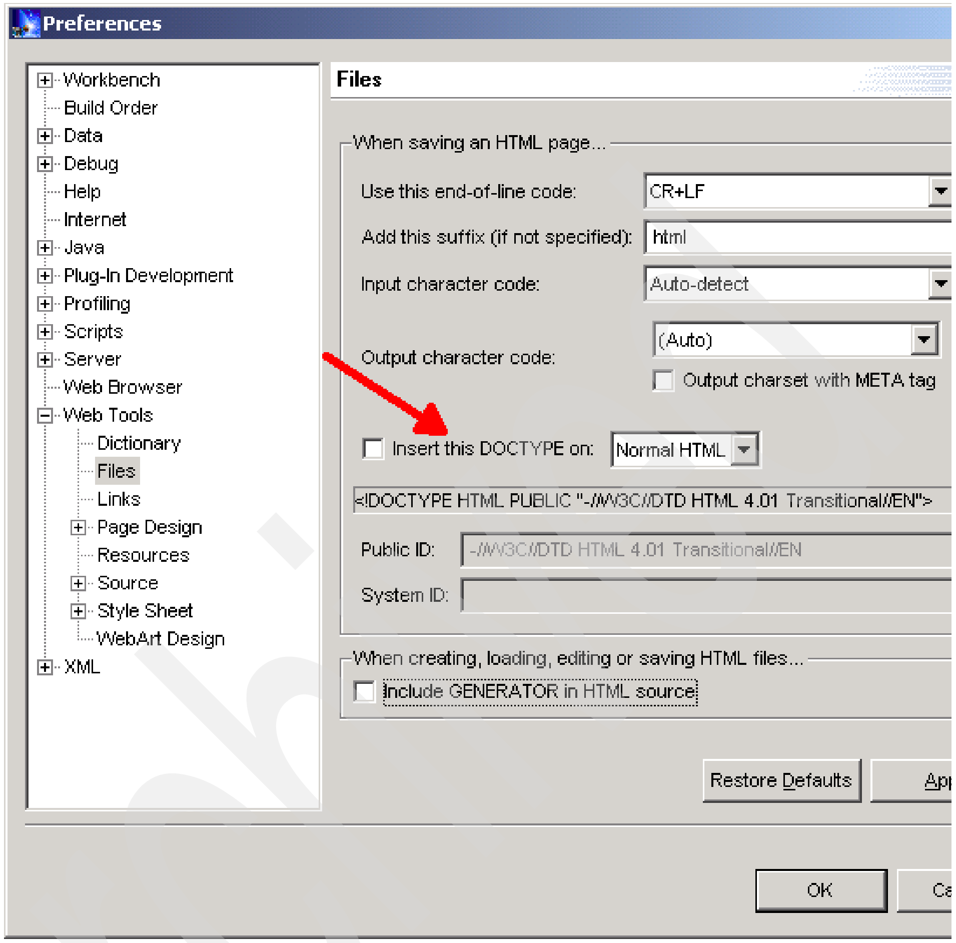
Figure 6-56 The newly renamed Logon.jsp

3) Changing JSPs

Next we need to modify the JSPs that were imported from the servlet. WebSphere Portal aggregates multiple portlet JSPs into a single Web page, so the portlet JSPs must consist only of page fragments. This means we need to remove all page level HTML tags from the JSPs, for example:

- <!DOCTYPE HTML - ....>
- <HTML>, </HTML>
- <HEAD>, </HEAD>
- <BODY>, </BODY>, including <META> and <LINK>
- <TITLE>, </TITLE>

Application Developer will automatically generate the DOCTYPE mark up for Web pages and JSPs and we needed to turn this off. Select **`Window`**->**`Preferences`**, expand **`Web Tools`** and select **`Files`**, as shown in Figure 6-57 on page 247.



Files

When saving an HTML page...

Use this end-of-line code: CR+LF

Add this suffix (if not specified): html

Input character code: Auto-detect

Output character code: (Auto)

Output charset with META tag

Insert this DOCTYPE on: Normal HTML

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

Public ID: -//W3C//DTD HTML 4.01 Transitional//EN

System ID:

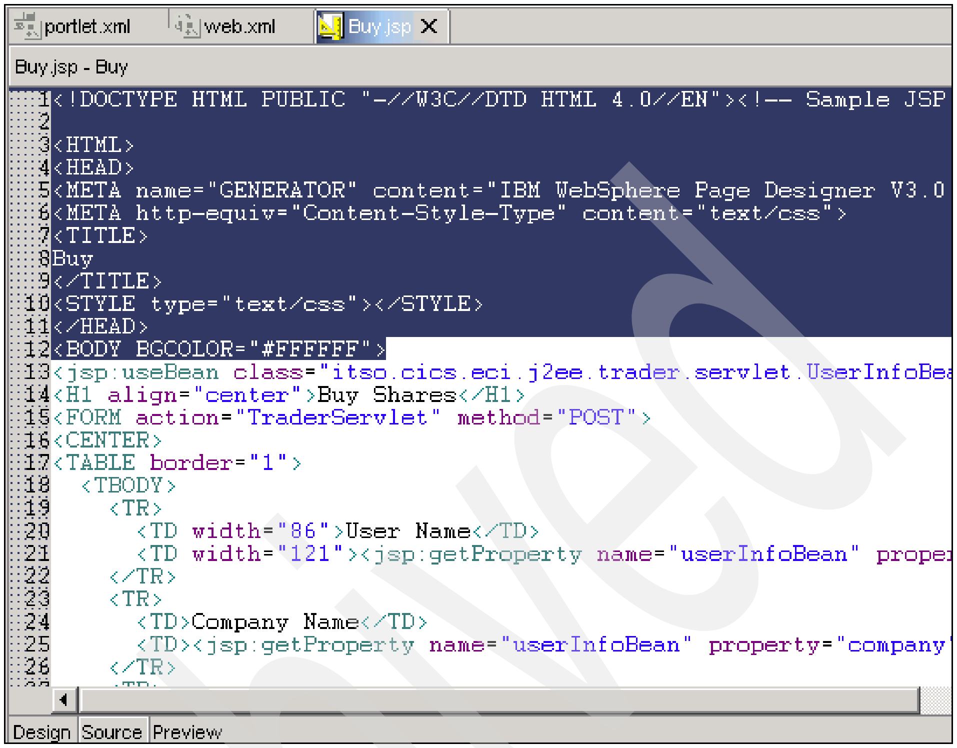
When creating, loading, editing or saving HTML files...

Include GENERATOR in HTML source

Figure 6-57 Remove insert DOCTYPE preference

Uncheck **Insert this DOCTYPE**, also **Include GENERATOR in HTML** source, and click **Apply** followed by **OK**.

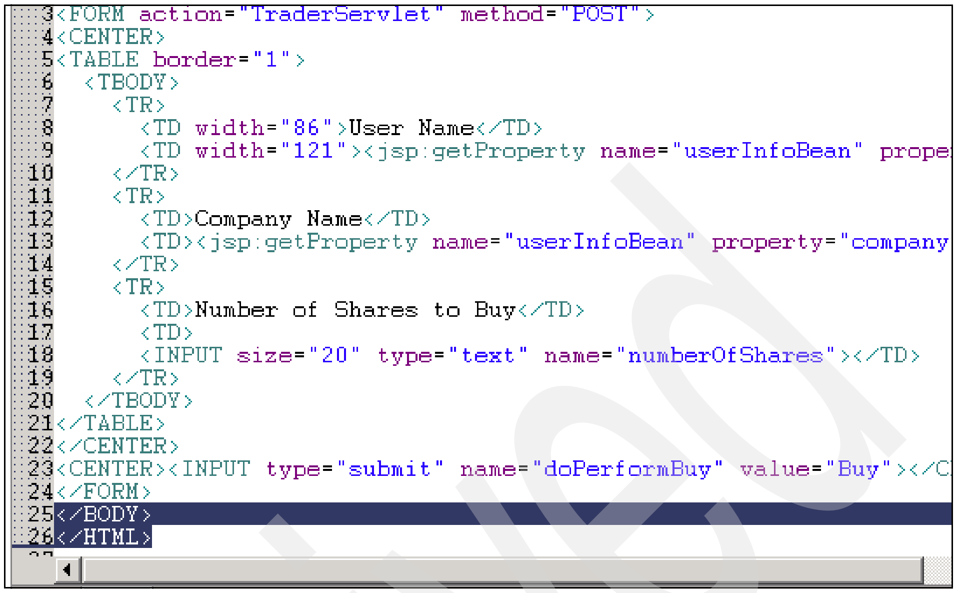
Next we modified JSPs `Buy.jsp`, `CompanySelection.jsp`, `Quotes.jsp`, `Sell.jsp` and `TraderError.jsp` to remove the page level tags. Using `Buy.jsp` as an example of how to do this we opened the source frame for the JSP and highlighted for deletion the tags above the <jsp:useBean...> tags as shown in Figure 6-58 on page 248.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"><!-- Sample JSP

<HTML>
<HEAD>
<META name="GENERATOR" content="IBM WebSphere Page Designer V3.0
<META http-equiv="Content-Style-Type" content="text/css">
<TITLE>
Buy
</TITLE>
<STYLE type="text/css"></STYLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<jsp:useBean class="itso.cics.eci.j2ee.trader.servlet.UserInfoBea
<H1 align="center">Buy Shares</H1>
<FORM action="TraderServlet" method="POST">
<CENTER>
<TABLE border="1">
  <TBODY>
    <TR>
      <TD width="86">User Name</TD>
      <TD width="121"><jsp:getProperty name="userInfoBean" proper
    </TR>
    <TR>
      <TD>Company Name</TD>
      <TD><jsp:getProperty name="userInfoBean" property="company"
    </TR>
```

Figure 6-58 deleting page level tags from the JSP, part 1

We left alone the <jsp:useBean...> tag and also other tags that form the data of the body of the JSP, and highlighted the remaining page level tag for deletion at the bottom of the JSP as seen in Figure 6-59 on page 249.



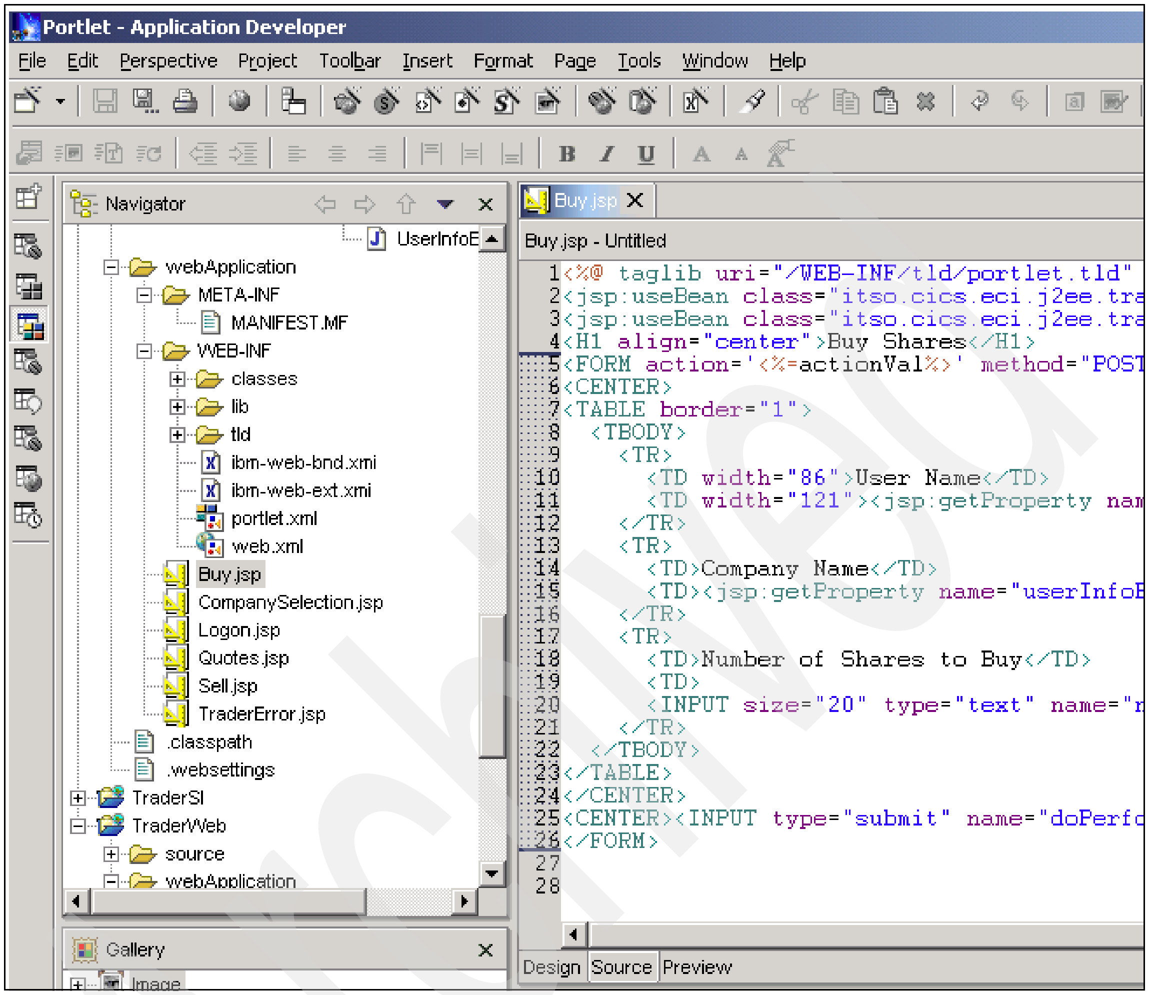
```
<FORM action="TraderServlet" method="POST">
<CENTER>
<TABLE border="1">
  <TBODY>
    <TR>
      <TD width="86">User Name</TD>
      <TD width="121"><jsp:getProperty name="userInfoBean" prope
    </TR>
    <TR>
      <TD>Company Name</TD>
      <TD><jsp:getProperty name="userInfoBean" property="company
    </TR>
    <TR>
      <TD>Number of Shares to Buy</TD>
      <TD>
      <INPUT size="20" type="text" name="numberOfShares"></TD>
    </TR>
  </TBODY>
</TABLE>
</CENTER>
<CENTER><INPUT type="submit" name="doPerformBuy" value="Buy"></C
</FORM>
</BODY>
</HTML>
```

Figure 6-59 deleting page level tags from the JSP, part 2

We next need to change the ACTION field in any FORM's in the JSPs. The action field in an HTML form is used to tell the browser which service, for example a Web page, servlet, CGI, and so on, needs to be used to perform the action associated with the form. The JSPs from the Trader Web application servlet had a hard coded relative URI value of `TraderServlet` that is used to by the browser to call the servlet. In the case of a portlet running under Portal Server we do not know until runtime the value of the service that will be allocated by portlet container, so the JSP needs to obtain the ACTION field value when it executes.

There are different ways that the ACTION field value can be obtained at runtime. We chose to set a parameter for this value inside the portlet code; see "5) Changing the servlet Java code to portlet code" on page 255, and extract it from the `PortletRequest` object using the following code in the `JSP`: `<FORM action='<%=request.getAttribute("actionVal")%>' method="POST">`. We made this change as appropriate to all of our JSPs:

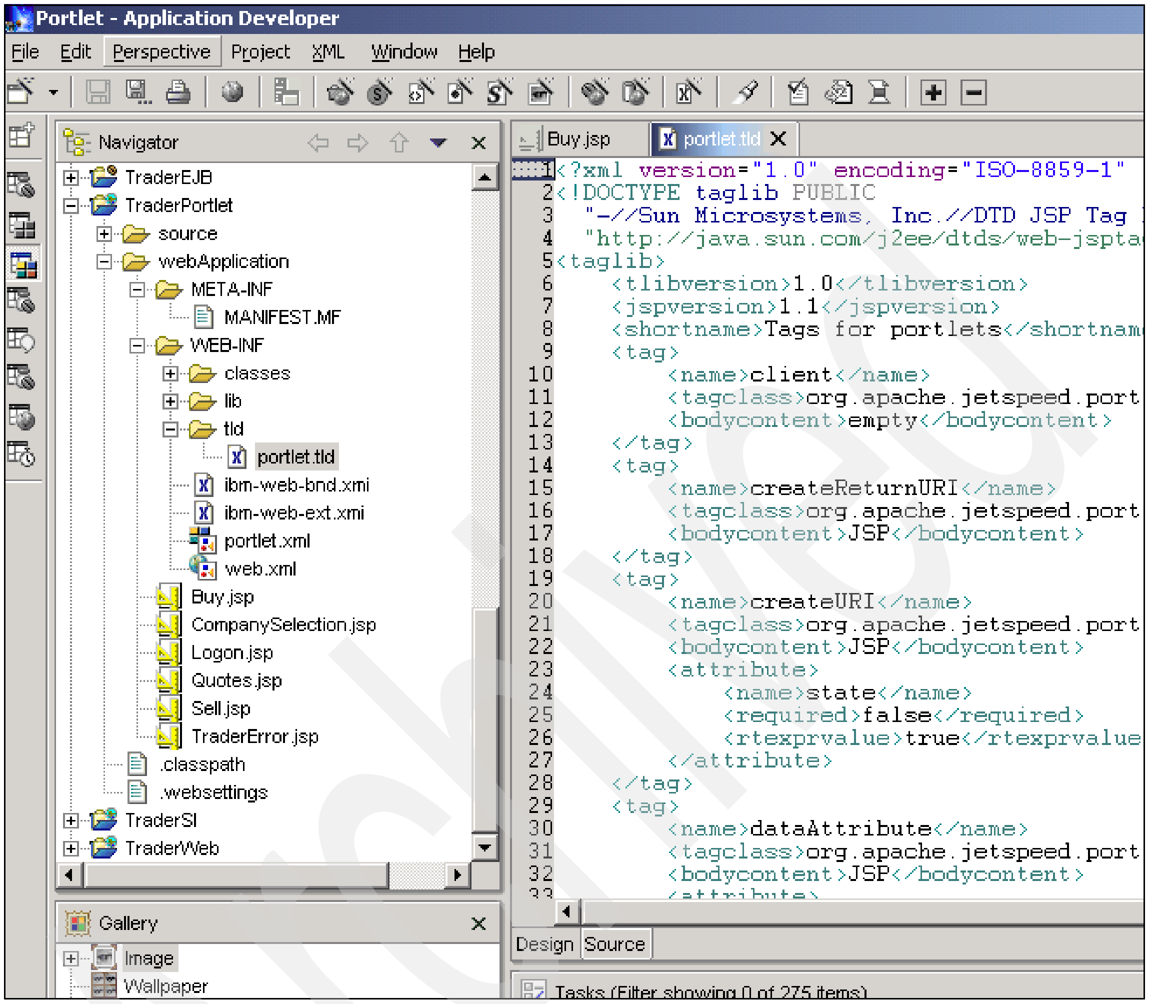
The next recommended step uses the Portal Server API from the portal tag library. This is done by adding statement `<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>` to the top of the JSPs, as shown in Figure 6-60 on page 250.


Buy.jsp - Untitled

```
<%@ taglib uri="/WEB-INF/tld/portlet.tld"
<jsp:useBean class="itso.cics.eci.j2ee.tra
<jsp:useBean class="itso.cics.eci.j2ee.tra
<H1 align="center">Buy Shares</H1>
<FORM action='<%=actionVal%>' method="POST
<CENTER>
<TABLE border="1">
  <TBODY>
    <TR>
      <TD width="86">User Name</TD>
      <TD width="121"><jsp:getProperty nam
    </TR>
    <TR>
      <TD>Company Name</TD>
      <TD><jsp:getProperty name="userInfoB
    </TR>
    <TR>
      <TD>Number of Shares to Buy</TD>
      <TD>
      <INPUT size="20" type="text" name="r
    </TR>
  </TBODY>
</TABLE>
</CENTER>
<CENTER><INPUT type="submit" name="doPerfo
</FORM>
```

Figure 6-60 Portal Tag library reference in Buy.jsp

Next add file `portlet.tld` to directory /WEB-INF in the project as shown in Figure 6-61 on page 251.



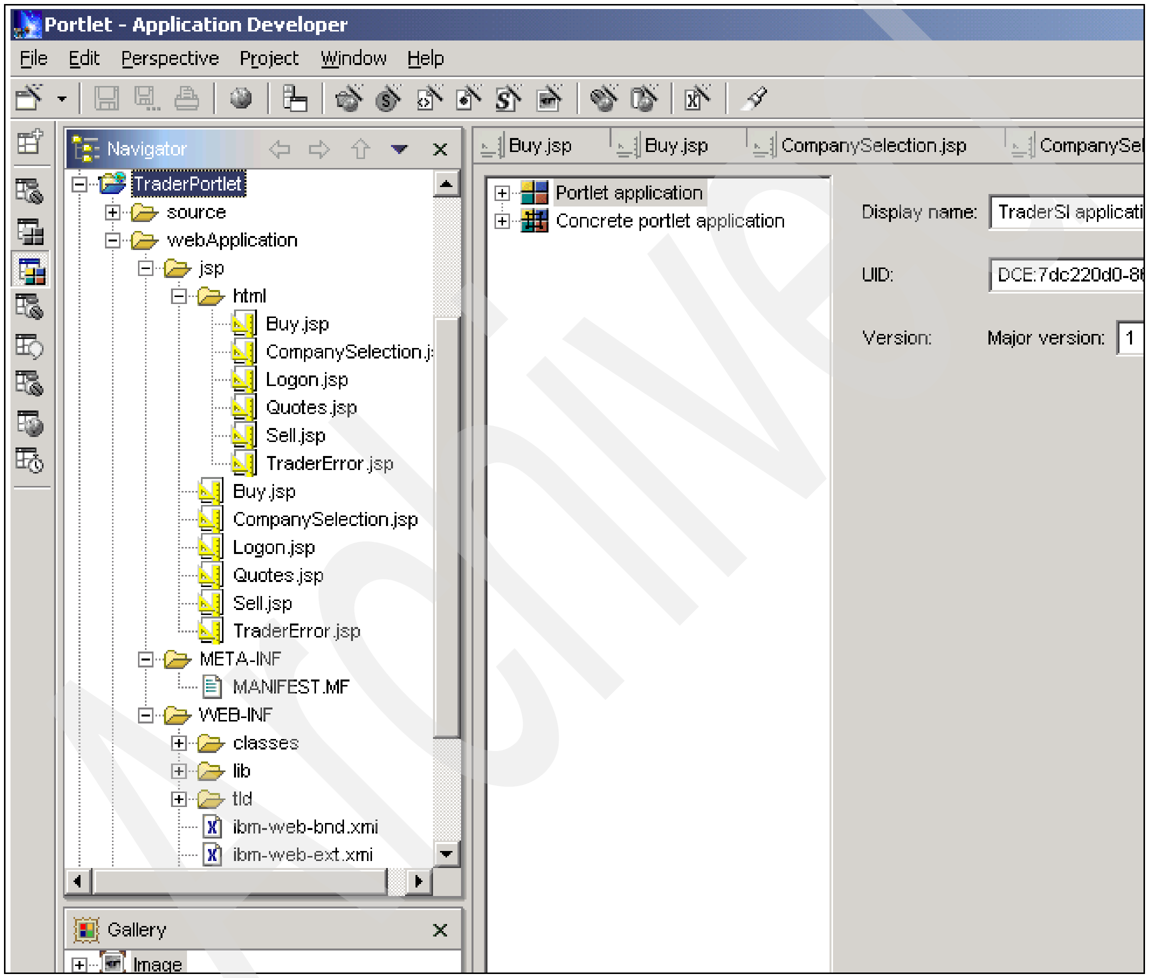
```
<?xml version="1.0" encoding="ISO-8859-1"
<!DOCTYPE taglib PUBLIC
  "-//Sun Microsystems, Inc.//DTD JSP Tag
  "http://java.sun.com/j2ee/dtds/web-jspta
<taglib>
    <tlibversion>1.0</tlibversion>
    <jspversion>1.1</jspversion>
    <shortname>Tags for portlets</shortnam
    <tag>
        <name>client</name>
        <tagclass>org.apache.jetspeed.port
        <bodycontent>empty</bodycontent>
    </tag>
    <tag>
        <name>createReturnURI</name>
        <tagclass>org.apache.jetspeed.port
        <bodycontent>JSP</bodycontent>
    </tag>
    <tag>
        <name>createURI</name>
        <tagclass>org.apache.jetspeed.port
        <bodycontent>JSP</bodycontent>
        <attribute>
            <name>state</name>
            <required>false</required>
            <rtexprvalue>true</rtexprvalue
        </attribute>
    </tag>
    <tag>
        <name>dataAttribute</name>
        <tagclass>org.apache.jetspeed.port
        <bodycontent>JSP</bodycontent>
```

Figure 6-61 Adding portlet.tld to the project directory

The portal tag library contains many useful functions that can be used by JSPs and we recommend that JSP and portlet developers become familiar with these functions. In our case we considered using the portal tag library to ensure that the INPUT fields on the FORMS inside are JSPs were assigned a unique value in the portal container namespace. This is necessary because the Portal Server reserves keyword values for its own use. An example of encoding an ACTION field in a form is to use a value such as:
`'<%= portletResponse.encodeNamespace("DisplayMode_attr")%>'`.
In the end we did not bother to do this as we were working with a small number of portlets, and the existing values from the servlet JSPs did not clash with any of

those reserved by the Portal Server, but it is the recommended practice for developing portlet JSPs.

The final change we made to the JSPs with our TraderPortlet project was to move the JSPs into project directory /WEB-INF, and additionally create another directory named /WEB-INF/html and copy the JSPs there as well as shown in Chapter 6, “JSPs in WEB-INF directory” on page 252.



Display name: TraderSI applicati

UID: DCE:7dc220d0-8

Version: Major version: 1

Figure 6-62   JSPs in WEB-INF directory

Although not strictly necessary for our small project, you should be aware that to support different mark-up languages and NLS language translations, the portlet container assumes there is a specific directory structure for them in directory WEB-INF, and searches them starting from the bottom of the directory tree.

Example 6-2 shows the original `CompanySelection.jsp` taken from the servlet where we have highlighted lines that are subject to modification or change based on the preceding comments.

Example 6-2 Original CompanySelection.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"><!-- Sample JSP file -->

<HTML>
<HEAD>
<META name="GENERATOR" content="IBM WebSphere Page Designer V3.0.2 for 
Windows">
<META http-equiv="Content-Style-Type" content="text/css">
<TITLE>
Company Selection
</TITLE>
</HEAD>

<BODY BGCOLOR="#FFFFFF">
<jsp:useBean class="itso.cics.eci.j2ee.trader.CompaniesBean" id="companiesBean" 
scope="request" />
<H1 align="center">Company Selection</H1>
<CENTER>
<TABLE width="500" border="1">
  <TBODY>
    <TR>
      <TD width="59" align="center" valign="middle"><B>Company</B></TD>
      <TD width="70" align="center" valign="middle"><B>Quotes</B></TD>
      <TD width="70" align="center" valign="middle"><B>Buy</B></TD>
      <TD width="68" align="center" valign="middle"><B>Sell</B></TD>
    </TR>
    <% for( int i=0; i<companiesBean.numOfCompanies();i++) {
         String compName = companiesBean.getCompany(i); %>
    <TR valign="top" align="center">
      <TD width="300" align="center" valign="top" height="27"><%= compName 
%></TD>
      <TD width="60" align="center" valign="middle" height="25">
      <FORM action="TraderServlet" method="POST"><INPUT type="submit" 
name="doShowQuotes" value="Quotes">
                 <INPUT type="hidden" name="company" value="<%= compName%>">
               </FORM>
      </TD>
      <TD width="60" align="center" valign="middle">
      <FORM action="TraderServlet" method="POST"><INPUT type="submit" 
name="doShowBuy" value="Buy">
                 <INPUT type="hidden" name="company" value="<%= compName%>">
               </FORM>
      </TD>
      <TD width="60" align="center" valign="middle">
      <FORM action="TraderServlet" method="POST"><INPUT type="submit"
name="doShowSell" value="Sell">
                 <INPUT type="hidden" name="company" value="<%= compName%>">
               </FORM>
      </TD>
    </TR>
    <% } %>
  </TBODY>
</TABLE>
</CENTER>
<FORM action="TraderServlet" method="POST">
<CENTER><INPUT type="submit" name="doShowLogoff" value="Logoff"></CENTER>
</FORM>
</BODY>
</HTML>
```

Example 6-3 shows `CompanySelection.jsp` from our new `TraderPortlet` portlet where we have highlighted the lines that we actually added or modified. Note that we added a `System.out.println` statement at one stage to help us with debug as noted in Chapter 4, "Log Files" on page 136.

Example 6-3 Modified CompanySelection.jsp

```
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<jsp:useBean class="itso.cics.eci.j2ee.trader.CompaniesBean" id="companiesBean"
scope="request" />

<H1 align="center">Company Selection</H1>
<CENTER>
<TABLE width="500" border="1">
  <TBODY>
    <TR>
      <TD width="59" align="center" valign="middle"><B>Company</B></TD>
      <TD width="70" align="center" valign="middle"><B>Quotes</B></TD>
      <TD width="70" align="center" valign="middle"><B>Buy</B></TD>
      <TD width="68" align="center" valign="middle"><B>Sell</B></TD>
    </TR>
    <% System.out.println("C= Henry Company JSP getAttribute = " +
(String)request.getAttribute("actionVal")); %>
    <% for( int i=0; i<companiesBean.numOfCompanies();i++) {
         String compName = companiesBean.getCompany(i); %>
    <TR valign="top" align="center">
      <TD width="300" align="center" valign="top" height="27"><%= compName
%></TD>
      <TD width="60" align="center" valign="middle" height="25">
      <% // Form action parameter obtained from portlet %>
      <FORM action='<%=request.getAttribute("actionVal")%>'
method="POST"><INPUT type="submit" name="doShowQuotes" value="Quotes">
```

```
                 <INPUT type="hidden" name="company" value="<%= compName%>">
               </FORM>
      </TD>
      <TD width="60" align="center" valign="middle">
      <% // Form action parameter obtained from portlet %>
      <FORM action='<%=request.getAttribute("actionVal")%>'
method="POST"><INPUT type="submit" name="doShowBuy" value="Buy">
                 <INPUT type="hidden" name="company" value="<%= compName%>">
               </FORM>
      </TD>
      <TD width="60" align="center" valign="middle">
      <% // Form action parameter obtained from portlet %>
      <FORM action='<%=request.getAttribute("actionVal")%>'
method="POST"><INPUT type="submit" name="doShowSell" value="Sell">
                 <INPUT type="hidden" name="company" value="<%= compName%>">
               </FORM>
      </TD>
    </TR>
    <% } %>
  </TBODY>
</TABLE>
</CENTER>
<% // Form action parameter obtained from portlet %>
<FORM action='<%=request.getAttribute("actionVal")%>' method="POST">
<CENTER><INPUT type="submit" name="doShowLogoff" value="Logoff"></CENTER>
</FORM>
```

4) Modifying web.xml

File `web.xml` from the servlet contained references to the welcome HTML page including `Logon.html` which we deleted. We also changed the url-pattern tag to `/TraderServlet/*`. Web.xml controls many of the execution and configuration parameters of the portlet -servlet, including for example security user ID's, that we did not explore here. For this project we could have left `web.xml` as it was, but for general conversion of a servlet to a portlet, examination and modification of this file will most likely be needed.

5) Changing the servlet Java code to portlet code

The major effort in changing the Trader Web application from a servlet to a portlet required changes to the servlet's Java code. The servlet consisted of two Java beans, `ErrorInfoBean.java` and `UserInfoBean.java`, that after examination we left unchanged.

The main servlet file `TraderServlet.java` needed quite significant modifications to convert it to a portlet. Some general items to be considered for this work are as follows.

1. Add the portlet API imports to the source, for example:
```
import org.apache.jetspeed.portlet.*;
import org.apache.jetspeed.portlets.*
```
2. Decide which portlet class to use for the implementation. In the simple Hello World example in 6.2, "Portlet development example" on page 185 we extended the `AbstractPortlet` class, but the `PortletAdapter` class provides a default implementation of the `AbstractPortlet` class, and is recommended instead.
3. Remove any doXXX() servlet methods, for example in our case `doGet()` and `doPost()`, and replace them with the `doView()` method instead. Consider adding other processing methods `doEdit()`, `doHelp()` and `doConfig()` to the portlet as appropriate.
4. Change servlet API calls to the HTTPSession object to use the portlet API instead. For example change:
```
UserInfoBean userInfo =
(UserInfoBean)httpSesion.getAttribute(userInfoID);
```
to
```
UserInfoBean userInfo =
(UserInfoBean)request.getAttribute(userInfoID);
```
5. Change other servlet API calls to use the equivalent portlet API calls, some examples:
   a. HttpServlet to PortletAdapter
   a. HttpServletRequest to PortletRequest
   b. HttpServletResponse to PortletResponse
   c. ServletException to PortletException
   d. HttpServletSession to PortletSession
6. Set JSPs FORM ACTION fields for runtime determination by JSPs, for example:
```
PortletURI traderURI = response.createURI();
PortletAction newAction = new DefaultPortletAction("logon");
traderURI.addAction(newAction);
request.setAttribute("actionVal", traderURI.toString());
```
7. Render the JSPs to display portlet views, for example:
```
getPortletConfig().getContext().include("/jsp/Logon.jsp", request,
response);
```
8. Add any state handling to the portlet for user interaction, for example to handle cases where users switch portal pages away from the portlet and the return again within the same session.

Example 6-4 shows the complete source code listing of the TraderServlet before we converted it into a TraderPortlet:

Example 6-4 The servlet code before modification

```
package itso.cics.eci.j2ee.trader.servlet;

import itso.cics.eci.j2ee.trader.*;
import java.io.*;
import javax.ejb.*;
import javax.servlet.http.*;
import javax.servlet.*;


public class TraderServlet extends javax.servlet.http.HttpServlet {

   // HttpSession IDs
   private final static String traderID = "Trader";
   private final static String companiesID = "Companies";
   private final static String userInfoID = "userInfo";



   // constants used in JSPs
   // field names
   private final static String fieldDoPerformLogon = "doPerformLogon";
   private final static String fieldDoShowQuotes = "doShowQuotes";
   private final static String fieldDoShowBuy = "doShowBuy";
   private final static String fieldDoPerformBuy = "doPerformBuy";
   private final static String fieldDoShowSell = "doShowSell";
   private final static String fieldDoPerformSell = "doPerformSell";
   private final static String fieldDoShowCompanies = "doShowCompanies";
   private final static String fieldDoShowLogoff = "doShowLogoff";

   private final static String fieldJndiName = "jndiName";
   private final static String fieldNameService = "nameService";
   private final static String fieldProviderURL = "providerURL";
   private final static String fieldUserID = "userid";
   private final static String fieldPassword = "password";
   private final static String fieldCompany = "company";
   private final static String fieldNumberOfShares = "numberOfShares";

   // JSPs
   private final static String jspLogon = "Logon.html";

   private final static String jspCompanySelection = "CompanySelection.jsp";
   private final static String jspQuotes = "Quotes.jsp";
   private final static String jspBuy = "Buy.jsp";
   private final static String jspSell = "Sell.jsp";
   private final static String jspTraderError = "TraderError.jsp";

   // beans
   private final static String beanCompanies = "companiesBean";
   private final static String beanQuotes = "quotesBean";
   private final static String beanUserInfo = "userInfoBean";
   private final static String beanErrorMessage = "errorMessageBean";



/**
 * Process incoming HTTP GET requests
 *
 * @param request Object that encapsulates the request to the servlet
 * @param response Object that encapsulates the response from the servlet
 */
public void doGet(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws
javax.servlet.ServletException, java.io.IOException {

   performTask(request, response);

}
/**
 * Process incoming HTTP POST requests
 *
 * @param request Object that encapsulates the request to the servlet
 * @param response Object that encapsulates the response from the servlet
 */
public void doPost(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws
javax.servlet.ServletException, java.io.IOException {

   performTask(request, response);

}
private String getCurrentCompany(HttpServletRequest request) throws Exception {

   HttpSession httpSesion = request.getSession();
   UserInfoBean userInfo = (UserInfoBean)httpSesion.getAttribute(userInfoID);
   return userInfo.getCompany();

}
public String getServletInfo() {

   return super.getServletInfo();

}
```

```
public String handleShowBuy(HttpServletRequest request) throws Exception {

   // query company and update it
   updateCompany(request);

   // return user info
   returnUserInfo(request);

   // show buy panel
   return jspBuy;

}

public String handleShowError(HttpServletRequest request, String errorText )
throws Exception {

   request.setAttribute(beanErrorMessage, new ErrorMessageBean(errorText));

   return jspTraderError;

}
public String handleShowError(HttpServletRequest request, String errorText,
String stackTrace ) throws Exception {

   request.setAttribute(beanErrorMessage, new ErrorMessageBean(errorText,
stackTrace));

   return jspTraderError;

}


public String handleShowSell(HttpServletRequest request) throws Exception {

   // query company and update it
   updateCompany(request);

   // return user info
   returnUserInfo(request);

   // show sell panel
   return jspSell;

}
/**
 * Initializes the servlet.
 */
```

```
public void init() {
   // insert code to initialize the servlet here

}
/**
 * Process incoming requests for information
 *
 * @param request Object that encapsulates the request to the servlet
 * @param response Object that encapsulates the response from the servlet
 */
public void performTask(HttpServletRequest request, HttpServletResponse
response) {

   try {
      // declare tader
      Trader trader = null;

      // obtain HttpSession
      HttpSession httpSesion = request.getSession();

      // if logon request
      if ( request.getParameter(fieldDoPerformLogon) != null ) {
         // create new trader session bean and keep it in session
         trader = createTrader(request);
         httpSesion.setAttribute(traderID, trader);
      }
      else {// for all other request trader must already exist
         // retrieve trader from session
         trader = (Trader)httpSesion.getAttribute(traderID);
      }

      // check which request we got and dispatch to appropriate method
      String nextJsp = null;

      if ( request.getParameter(fieldDoPerformLogon) != null )
         nextJsp = handlePerformLogon(request, trader);
      else if ( request.getParameter(fieldDoShowQuotes) != null )
         nextJsp = handleShowQuotes(request, trader);
      else if ( request.getParameter(fieldDoShowCompanies) != null )
         nextJsp = handleShowCompanies(request, trader);
      else if ( request.getParameter(fieldDoShowBuy) != null )
         nextJsp = handleShowBuy(request);
      else if ( request.getParameter(fieldDoPerformBuy) != null )
         nextJsp = handlePerformBuy(request, trader);
      else if ( request.getParameter(fieldDoShowSell) != null )
         nextJsp = handleShowSell(request);
      else if ( request.getParameter(fieldDoPerformSell) != null )
         nextJsp = handlePerformSell(request, trader);
      else if ( request.getParameter(fieldDoShowLogoff) != null )
```

```
         nextJsp = handleShowLogoff(request, trader);
      else
         nextJsp = handleShowError(request, "Got unknown request to process, 
check the JSPs.");

      // now process JSP
      ServletContext sc = getServletContext();
      RequestDispatcher rd = sc.getRequestDispatcher("/" + nextJsp);
      rd.forward(request, response);

   }
   catch(Throwable theException) {
      try {
         trace( "We got an exception in TraderServlet.performTask" );
         theException.printStackTrace();

         ByteArrayOutputStream bas = new ByteArrayOutputStream();
         PrintStream ps = new PrintStream( bas );
         theException.printStackTrace(ps);

         String nextJsp = handleShowError(request, "Error processing bean: " + 
theException.toString(), bas.toString() );

         ServletContext sc = getServletContext();
         RequestDispatcher rd = sc.getRequestDispatcher("/" + nextJsp);
         rd.forward(request, response);
      }
      catch( Throwable t ) {
         System.out.println( "Fatal error, exception in exception handling!" 
);
         t.printStackTrace();
      }
   }

}
private void returnUserInfo(HttpServletRequest request) throws Exception {

   // return user info
   HttpSession httpSesion = request.getSession();
   UserInfoBean userInfo = (UserInfoBean)httpSesion.getAttribute(userInfoID);
   request.setAttribute(beanUserInfo, userInfo);

}
private void trace(String txt) {
   System.err.println( txt );
}
private void updateCompany(HttpServletRequest request) {

   String company = request.getParameter(fieldCompany);
```

```
    HttpSession httpSesion = request.getSession();
    UserInfoBean userInfo = (UserInfoBean)httpSesion.getAttribute(userInfoID);
    userInfo.setCompany(company);
    request.getSession().setAttribute(userInfoID, userInfo );

}
private Trader createTrader(HttpServletRequest request) throws
javax.naming.NamingException, javax.ejb.CreateException,
java.rmi.RemoteException {

    // retrieve fields
    String jndiName = request.getParameter(fieldJndiName);
    String nameService = request.getParameter(fieldNameService);
    String providerURL = request.getParameter(fieldProviderURL);

    // create initial context
    java.util.Hashtable properties = new java.util.Hashtable(2);
    properties.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY, nameService);
    properties.put(javax.naming.Context.PROVIDER_URL, providerURL);

    javax.naming.InitialContext ctx = new
javax.naming.InitialContext(properties);

    // lookup TraderHome
    Object obj = ctx.lookup(jndiName); // Use the JNDI name from HTML page

    // narrow to TraderHome and create Trader
    TraderHome traderHome =
(TraderHome)javax.rmi.PortableRemoteObject.narrow((org.omg.CORBA.Object)obj,
TraderHome.class);
    Trader trader = traderHome.create();

    return trader;
}public String handlePerformBuy(HttpServletRequest request, Trader trader)
throws Exception {

    // get the number of shares
    String numberOfShares = request.getParameter(fieldNumberOfShares);
    String company = getCurrentCompany(request);

    // convert to integer
    int shares = 0;
    try {
        shares = Integer.parseInt(numberOfShares);
    } catch( Exception e ) {
    }

    // buy shares
    trader.buy( company, shares );
```

```
   // return user info
   returnUserInfo(request);

   // go back to companies selction
   return handleShowCompanies(request, trader);

}public String handlePerformLogon(HttpServletRequest request, Trader trader )
throws Exception {

   // retrieve fields
   String userID = request.getParameter(fieldUserID);
   String password = request.getParameter(fieldPassword);

   // check if userID and password are provided
   if( userID.equals("") || password.equals("") )
      return handleShowError(request, "You have to specify userID AND
password" );

   // now logon to the application
   trader.logon( userID, password);

   // store user info, we need it later
   UserInfoBean userInfo = new UserInfoBean();
   userInfo.setUserID(userID);
   request.getSession().setAttribute(userInfoID, userInfo );

   // show companies now
   return handleShowCompanies(request, trader);

}public String handlePerformSell(HttpServletRequest request, Trader trader)
throws Exception {

   // get the number of shares
   String numberOfShares = request.getParameter(fieldNumberOfShares);
   String company = getCurrentCompany(request);

   // convert to integer
   int shares = 0;
   try {
      shares = Integer.parseInt(numberOfShares);
   } catch( Exception e ) {
   }

   // sell shares
   trader.sell( company, shares );

   // return user info
   returnUserInfo(request);
   // go back to companies selction
   return handleShowCompanies(request,trader);

}public String handleShowCompanies(HttpServletRequest request, Trader trader)
throws Exception {

   // check for local copy
   HttpSession httpSesion = request.getSession();
   CompaniesBean companies =
(CompaniesBean)httpSesion.getAttribute(companiesID);

   if( companies == null ) {// if not stored locally
      companies = trader.getCompanies();

      // to improve response time, store it locally
      request.getSession().setAttribute(companiesID, companies );
   }

   request.setAttribute(beanCompanies, companies);

   return jspCompanySelection;
}public String handleShowLogoff(HttpServletRequest request, Trader trader)
throws Exception {

   // logoff from trader
   trader.logoff();

   // remove trader object in EJB server
   trader.remove();

   // remove session variables
   request.getSession().removeAttribute(traderID);
   request.getSession().removeAttribute(companiesID);
   request.getSession().removeAttribute(userInfoID);

   // return to logon panel
   return jspLogon;

}public String handleShowQuotes(HttpServletRequest request, Trader trader)
throws Exception {

   // query company and update it
   updateCompany(request);

   // get the quotes
   String company = request.getParameter(fieldCompany);
   QuotesBean quotes = trader.getQuotes(company);
   // return quotes
   request.setAttribute(beanQuotes, quotes);

   // return user info
   returnUserInfo(request);

   // show the quotes
   return jspQuotes;

}
}
```

Example 6-5 shows the complete source code listing of new TraderPortlet modified from the original servlet where we have highlighted new or modified lines. Note that we tried to leave the code logic intact but have re-ordered the methods to a more meaningful order (init, doView, performTask, and so on):

Example 6-5 The Portlet code after modification

```
package itso.cics.eci.j2ee.trader.servlet;

import itso.cics.eci.j2ee.trader.*;
import java.io.*;
import javax.ejb.*;

// Added for portlet
import org.apache.jetspeed.portlet.*;
import org.apache.jetspeed.portlets.*;

/**
 * TraderServlet converted to run as a portlet
 */
public class TraderServlet extends org.apache.jetspeed.portlet.PortletAdapter {

   // PortletSession IDs
   private final static String traderID = "Trader";
   private final static String companiesID = "Companies";
   private final static String userInfoID = "userInfo";

   // constants used in JSPs
   // field names
   private final static String fieldDoPerformLogon = "doPerformLogon";
   private final static String fieldDoShowQuotes = "doShowQuotes";
   private final static String fieldDoShowBuy = "doShowBuy";
   private final static String fieldDoPerformBuy = "doPerformBuy";
   private final static String fieldDoShowSell = "doShowSell";
   private final static String fieldDoPerformSell = "doPerformSell";
```

```
    private final static String fieldDoShowCompanies = "doShowCompanies";
    private final static String fieldDoShowLogoff = "doShowLogoff";

    private final static String fieldJndiName = "jndiName";
    private final static String fieldNameService = "nameService";
    private final static String fieldProviderURL = "providerURL";
    private final static String fieldUserID = "userid";
    private final static String fieldPassword = "password";
    private final static String fieldCompany = "company";
    private final static String fieldNumberOfShares = "numberOfShares";
    private String state = null;

    // JSPs
    private final static String jspLogon = "Logon.jsp";

    private final static String jspCompanySelection = "CompanySelection.jsp";
    private final static String jspQuotes = "Quotes.jsp";
    private final static String jspBuy = "Buy.jsp";
    private final static String jspSell = "Sell.jsp";
    private final static String jspTraderError = "TraderError.jsp";

    // beans
    private final static String beanCompanies = "companiesBean";
    private final static String beanQuotes = "quotesBean";
    private final static String beanUserInfo = "userInfoBean";
    private final static String beanErrorMessage = "errorMessageBean";


/**
 * Initializes the portlet.
 */
public void init(PortletConfig portletConfig) throws UnavailableException {

      super.init(portletConfig);
      plog("TraderPortlet - init()");

}

/**
 * doView method added for the portlet, (instead of doGet, doPost)
 */
public void doView(PortletRequest request, PortletResponse response) throws
PortletException, java.io.IOException {

      // Set state for processing JSP's
      state = (String)request.getPortletSession().getAttribute("state");
      plog("TraderPortlet - doView(), state = " + state);

      // If first time through? update state, show Logon JSP
      if (state == null) {
         state = "logon";
         request.getPortletSession().setAttribute("state", state);

          // Create URI for the logon JSP FORM action attribute
         PortletURI traderURI = response.createURI();
         PortletAction newAction = new DefaultPortletAction("logon");
       traderURI.addAction(newAction);

       // save the URI so the JSP can get it
         request.setAttribute("actionVal", traderURI.toString());

         // Display logon JSP to render
       getPortletConfig().getContext().include("/jsp/Logon.jsp", request,
response);
      }

      // Not first time thorough, process task(s) from JSP
      else {
          performTask(request, response);
      }
}


/**
 * Process incoming requests for information
 * (session handling changed for portlet and state tracking)
 */
public void performTask(PortletRequest request, PortletResponse response) {

   plog("TraderPortlet - performTask()");

   try {
      // declare tader
      Trader trader = null;

      // if logon request
      if ( request.getParameter(fieldDoPerformLogon) != null ) {

         // create new trader session bean and keep it in session
         trader = createTrader(request);
         request.getPortletSession().setAttribute(traderID, trader);
      }
      else {
      // for all other request trader must already exist

         // so retrieve trader from session
         trader = (Trader)request.getPortletSession().getAttribute(traderID);
      }
```

```
         // check which request we got and dispatch to appropriate method
         String nextJsp = null;

         if ( request.getParameter(fieldDoPerformLogon) != null ){
            nextJsp = handlePerformLogon(request, trader);
            request.getPortletSession().setAttribute("state",
   fieldDoPerformLogon);
         }
         else if ( request.getParameter(fieldDoShowQuotes) != null ){
            nextJsp = handleShowQuotes(request, trader);
            request.getPortletSession().setAttribute("state", fieldDoShowQuotes);
         }
         else if ( request.getParameter(fieldDoShowCompanies) != null ){
            nextJsp = handleShowCompanies(request, trader);
            request.getPortletSession().setAttribute("state",
   fieldDoShowCompanies);
         }
         else if ( request.getParameter(fieldDoShowBuy) != null ){
            nextJsp = handleShowBuy(request);
            request.getPortletSession().setAttribute("state", fieldDoShowBuy);
         }
         else if ( request.getParameter(fieldDoPerformBuy) != null ){
            nextJsp = handlePerformBuy(request, trader);
            request.getPortletSession().setAttribute("state", fieldDoPerformBuy);
         }
         else if ( request.getParameter(fieldDoShowSell) != null ){
            nextJsp = handleShowSell(request);
            request.getPortletSession().setAttribute("state", fieldDoShowSell);
         }
         else if ( request.getParameter(fieldDoPerformSell) != null ){
            nextJsp = handlePerformSell(request, trader);
            request.getPortletSession().setAttribute("state",
   fieldDoPerformSell);
         }
         else if ( request.getParameter(fieldDoShowLogoff) != null ){
            nextJsp = handleShowLogoff(request, trader);
            request.getPortletSession().setAttribute("state", fieldDoShowLogoff);
         }
         // unknown request, try to get back to where we were
         else {
            state = (String)request.getPortletSession().getAttribute("state");

            if (state.equals(fieldDoPerformLogon)) {
            nextJsp = handlePerformLogon(request, trader);
            }
            else if (state.equals(fieldDoShowQuotes)) {
            nextJsp = handleShowQuotes(request, trader);
            }
            else if (state.equals(fieldDoShowCompanies)) {
            nextJsp = handleShowCompanies(request, trader);
            }
            else if (state.equals(fieldDoShowBuy)) {
            nextJsp = handleShowBuy(request);
            }
            else if (state.equals(fieldDoPerformBuy)) {
            nextJsp = handlePerformBuy(request, trader);
            }
            else if (state.equals(fieldDoShowSell)) {
            nextJsp = handleShowSell(request);
            }
            else if (state.equals(fieldDoPerformSell)) {
            nextJsp = handlePerformSell(request, trader);
            }
            else if (state.equals(fieldDoShowLogoff)) {
            nextJsp = handleShowLogoff(request, trader);
            }
        }

        plog("TraderPortlet in PerformTask(), nextJSP = " + nextJsp);
        // Changed for portlet
        // Create URI for the logon JSP FORM action attribute
        PortletURI traderURI = response.createURI();
        PortletAction newAction = new DefaultPortletAction("company");
        traderURI.addAction(newAction);

        // save the URI so the JSP can get it
        request.setAttribute("actionVal", traderURI.toString());

        // Display next JSP to render
         getPortletConfig().getContext().include("/jsp/" + nextJsp, request,
response);

    }
    catch(Throwable theException) {
        try {
            trace( "WebSphere Portal : we got an exception in
TraderPortlet.performTask" );
            theException.printStackTrace();

            ByteArrayOutputStream bas = new ByteArrayOutputStream();
            PrintStream ps = new PrintStream( bas );
            theException.printStackTrace(ps);

            String nextJsp = handleShowError(request, "Error processing bean: " +
theException.toString(), bas.toString() );

            // Changed for portlet
            /// Display next JSP to render
         getPortletConfig().getContext().include("/jsp/" + nextJsp, request,
response);
        }
        catch( Throwable t ) {

            t.printStackTrace();
        }
    }

}

/**
 * Extract current Company from UserInfoBean.
 */
private String getCurrentCompany(PortletRequest request) throws Exception {


    plog("TraderPortlet - getCurrentCompany()");
    // Session access changed for portlet
    UserInfoBean userInfo =
(UserInfoBean)request.getPortletSession().getAttribute(userInfoID);
    return userInfo.getCompany();

}

/**
 * Process buy request
 */
public String handleShowBuy(PortletRequest request) throws Exception {

    plog("TraderPortlet - handleShowBuy()");
    // query company and update it
    updateCompany(request);

    // return user info
    returnUserInfo(request);

    // show buy panel
    return jspBuy;

}

/**
 * Process an error
 */
public String handleShowError(PortletRequest request, String errorText ) throws
Exception {
   request.setAttribute(beanErrorMessage, new ErrorMessageBean(errorText));

   return jspTraderError;

}

/**
 * Process an error with Stack
 */
public String handleShowError(PortletRequest request, String errorText, String
stackTrace ) throws Exception {

   request.setAttribute(beanErrorMessage, new ErrorMessageBean(errorText,
stackTrace));

   return jspTraderError;

}

/**
 * Process sell request
 */
public String handleShowSell(PortletRequest request) throws Exception {


   plog("TraderPortlet - handleShowSell()");
   // query company and update it
   updateCompany(request);

   // return user info
   returnUserInfo(request);

   // show sell panel
   return jspSell;

}

/**
 * Get user info and set in UserInfoBean
 */
private void returnUserInfo(PortletRequest request) throws Exception {

   plog("TraderPortlet - returnUserInfo()");
   // Session handling changed for portlet, get bean from session
   UserInfoBean userInfo =
(UserInfoBean)request.getPortletSession().getAttribute(userInfoID);

   //set UserInfoBean for JSP
   request.setAttribute(beanUserInfo, userInfo);
```

```
}


/**
 * Print exceptions in error log
 */
private void trace(String txt) {

   System.err.println( txt );
}

/**
 * Added for portlet, log data to Portal Log
 */
private void plog(String txt) {

   PortletLog log = getPortletConfig().getContext().getLog();
   if (log.isDebugEnabled()) log.debug(txt);
   if (log.isWarnEnabled()) log.warn(txt);
   if (log.isInfoEnabled()) log.info(txt);
   if (log.isErrorEnabled()) log.error(txt);
}


/**
 * Get company selected and save in UserInfoBean
 */
private void updateCompany(PortletRequest request) {

   plog("TraderPortlet - updateCompany()");

   // Get company selected by user
   String company = request.getParameter(fieldCompany);

   // Session handling changed for portlet
   UserInfoBean userInfo =
(UserInfoBean)request.getPortletSession().getAttribute(userInfoID);

   // save in UserInfoBean
   userInfo.setCompany(company);

   // save in session for JSP
   request.getPortletSession().setAttribute(userInfoID, userInfo );

}


/**
```

```
 * Lookup and accesss Trader EJB
 */
private Trader createTrader(PortletRequest request) throws
javax.naming.NamingException, javax.ejb.CreateException,
java.rmi.RemoteException {


   // Retrieve fields from logon page
   String jndiName = request.getParameter(fieldJndiName);
   String nameService = request.getParameter(fieldNameService);
   String providerURL = request.getParameter(fieldProviderURL);

   // create initial context
   java.util.Hashtable properties = new java.util.Hashtable(2);
   properties.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY, nameService);
   properties.put(javax.naming.Context.PROVIDER_URL, providerURL);

   javax.naming.InitialContext ctx = new
javax.naming.InitialContext(properties);

   Object obj = ctx.lookup(jndiName); // Use the JNDI name from logon JSP

   // narrow to TraderHome and create Trader
   TraderHome traderHome =
(TraderHome)javax.rmi.PortableRemoteObject.narrow((org.omg.CORBA.Object)obj,
TraderHome.class);
   Trader trader = traderHome.create();

   return trader;
}

/**
 * Process buy request
 */
public String handlePerformBuy(PortletRequest request, Trader trader) throws
Exception {


   plog("TraderPortlet - handlePerformBuy()");
   // get the number of shares
   String numberOfShares = request.getParameter(fieldNumberOfShares);
   String company = getCurrentCompany(request);

   // convert to integer
   int shares = 0;
   try {
      shares = Integer.parseInt(numberOfShares);
   }
   catch( Exception e ) {
    }

    // buy shares
    trader.buy( company, shares );

    // return user info
    returnUserInfo(request);

    // go back to companies selection
    return handleShowCompanies(request, trader);

}

/**
 * Process logon request
 */
public String handlePerformLogon(PortletRequest request, Trader trader ) throws
Exception {


    plog("TraderPortlet - handlePerformLogon()");
    // retrieve input fields user and password
    String userID = request.getParameter(fieldUserID);
    String password = request.getParameter(fieldPassword);

    // check if userID and password are provided
    if( userID.equals("") || password.equals("") )
        return handleShowError(request, "You have to specify userID AND
password" );

    // now logon to the application
    trader.logon( userID, password);

    // store user info, we need it later
    UserInfoBean userInfo = new UserInfoBean();
    userInfo.setUserID(userID);
    // session changed for portlet
    request.getPortletSession().setAttribute(userInfoID, userInfo );

    // show companies now
    return handleShowCompanies(request, trader);

}

/**
 * Process sell request
 */
public String handlePerformSell(PortletRequest request, Trader trader) throws
Exception {
    plog("TraderPortlet - handlePerformSell()");
    // get the number of shares
    String numberOfShares = request.getParameter(fieldNumberOfShares);
    String company = getCurrentCompany(request);

    // convert to integer
    int shares = 0;
    try {
        shares = Integer.parseInt(numberOfShares);
    } catch( Exception e ) {
    }

    // sell shares
    trader.sell( company, shares );

    // return user info
    returnUserInfo(request);

    // go back to companies selection
    return handleShowCompanies(request,trader);

}

/**
 * Show companies list
 */
public String handleShowCompanies(PortletRequest request, Trader trader) throws
Exception {

    plog("TraderPortlet - handleShowCompanies()");
    // check for local copy of bean (session handling changed for portlet)
    CompaniesBean companies =
(CompaniesBean)request.getPortletSession().getAttribute(companiesID);

    // if not stored locally in session
    if( companies == null ) {
        companies = trader.getCompanies();

        // to improve response time, store it locally
        request.getPortletSession().setAttribute(companiesID, companies );
    }

     // set bean for JSP
    request.setAttribute(beanCompanies, companies);

    return jspCompanySelection;
}
```

```
/**
 * Handle logoff
 */
public String handleShowLogoff(PortletRequest request, Trader trader) throws
Exception {

   plog("TraderPortlet - handleShowLogoff()");
   // logoff from trader
   trader.logoff();

   // remove trader object in EJB server
   trader.remove();

   // remove session variables (session handling changed for portlet)
   request.getPortletSession().removeAttribute(traderID);
   request.getPortletSession().removeAttribute(companiesID);
   request.getPortletSession().removeAttribute(userInfoID);

   // return to logon panel
   return jspLogon;

}

/**
 * Show requested quotes
 */
public String handleShowQuotes(PortletRequest request, Trader trader) throws
Exception {


   plog("TraderPortlet - handleShowQuotes()");
   // query company and update it
   updateCompany(request);

   // get the company requested
   String company = request.getParameter(fieldCompany);
   // and get the quotes
   QuotesBean quotes = trader.getQuotes(company);

   // return quotes
   request.setAttribute(beanQuotes, quotes);

   // return user info
   returnUserInfo(request);

   // show the quotes
   return jspQuotes;

}
```

```
}
```

---

6) Other things to consider when converting a servlet to a portlet

We have already covered the first four of the following items but list them here for convenience:

1. Add the Portal API available from the WebSphere Portal tag library
   `<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>`
   then the JSP can use the Portal API. An example of this is shown in the next item to define a unique form namespace. If the JSP uses the Portal tag library then the library must be available in the WAR file directory /WEB-INF/tld.
2. Ensure the HTML object namespace is unique within WebSphere Portal
   We did not do this, but an example applied to the INPUT field of the form in `Buy.jsp` is:
   ```
   <INPUT type="submit" name='<portletAPI:encodeNamespace
   value="Buy"/>'>
   ```
3. Set the action attribute of the forms in the JSPs from the `doView`, `doPost` and `doGet` methods in the portlet:
   ```
   PortletURI traderURI = response.createURI();
   request.setAttribute("actionVal", traderURI.toString());
   ```
   Then in the JSP form the attribute must be evaluated, for example:
   `<FORM action='<%=actionVal%>'>`
4. In the JSPs with forms, retrieve the URI for the action attribute:
   ```
   <jsp:useBean class="itso.cics.eci.j2ee.trader.servlet.UserInfoBean"
   id="actionVal" scope="request" />
   ```
5. Add desired Portal themes and skins to the portlet
   We did not do this.
6. Cleanup file names and variables to remove references to `servlet`
   We did not do this.
7. Add additional mark-up language support to the portlet
   We did not do this.
8. Import an EJB application client JAR if the portlet references EJBs
   For the Trader application, as we have seen in Figure 6-34 on page 224 the Web application part of the servlet, and therefore the TraderPortlet java code, performs a lookup of the JNDI namespace for the EJB home of the session bean used to wrapper the EJB that connects to CICs. At this stage this was reflected in the many cross reference errors we were seeing in the Application Developer tasks list when building the portal project.

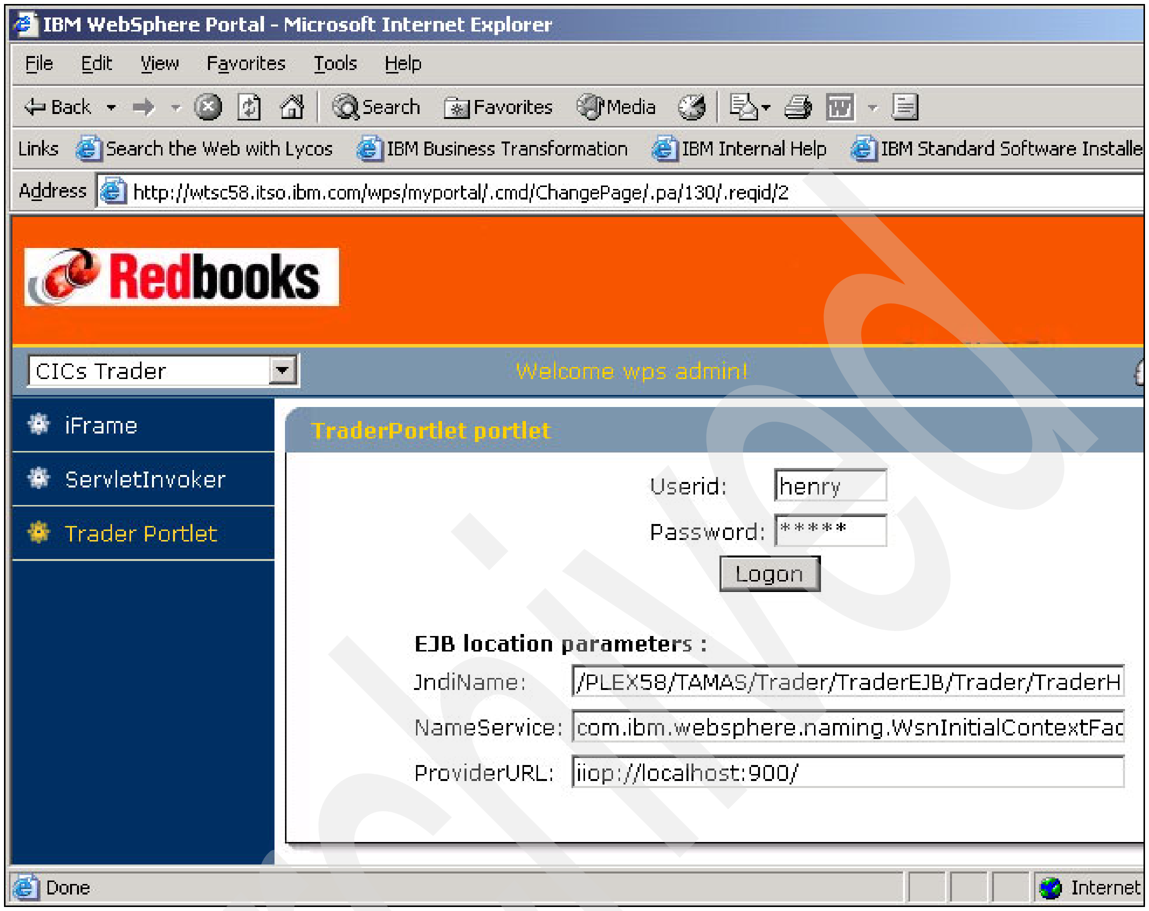
   From the portlet perspective of the TraderEJB project in Application Developer we identified file `imported_classes_EJB.jar` which we found in the

Application Developer directory structure file-system, and subsequently imported into the /lib directory of TraderPortlet project. This immediately resolved all of our build errors caused by missing references to the EJB.

Tip: Additional information on portlet development is available in redbook *IBM WebSphere Portal Developers Handbook,* SG24-6897*.*

At this stage with building the portlet project, assuming there are no errors, listed by Application Developer as tasks, it is ready for either debug and testing using the development environment we described in Chapter 3, “Installing the Portal Toolkit” on page 104, or deployment directly to WebSphere Portal on z/OS for testing.

After deploying TraderServlet to WebSphere Portal, and adding it to a page we had created in out of CICS’ test application place, we used the logon page rendered from our Logon.jsp and shown in Figure 6-63 on page 279 to sign on to the Trader application.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installe
Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ChangePage/.pa/130/.reqid/2
Redbooks
CICs Trader
Welcome wps admin!
iFrame
ServletInvoker
Trader Portlet
TraderPortlet portlet
Logon
Done
Internet

Userid: henry
Password: *****

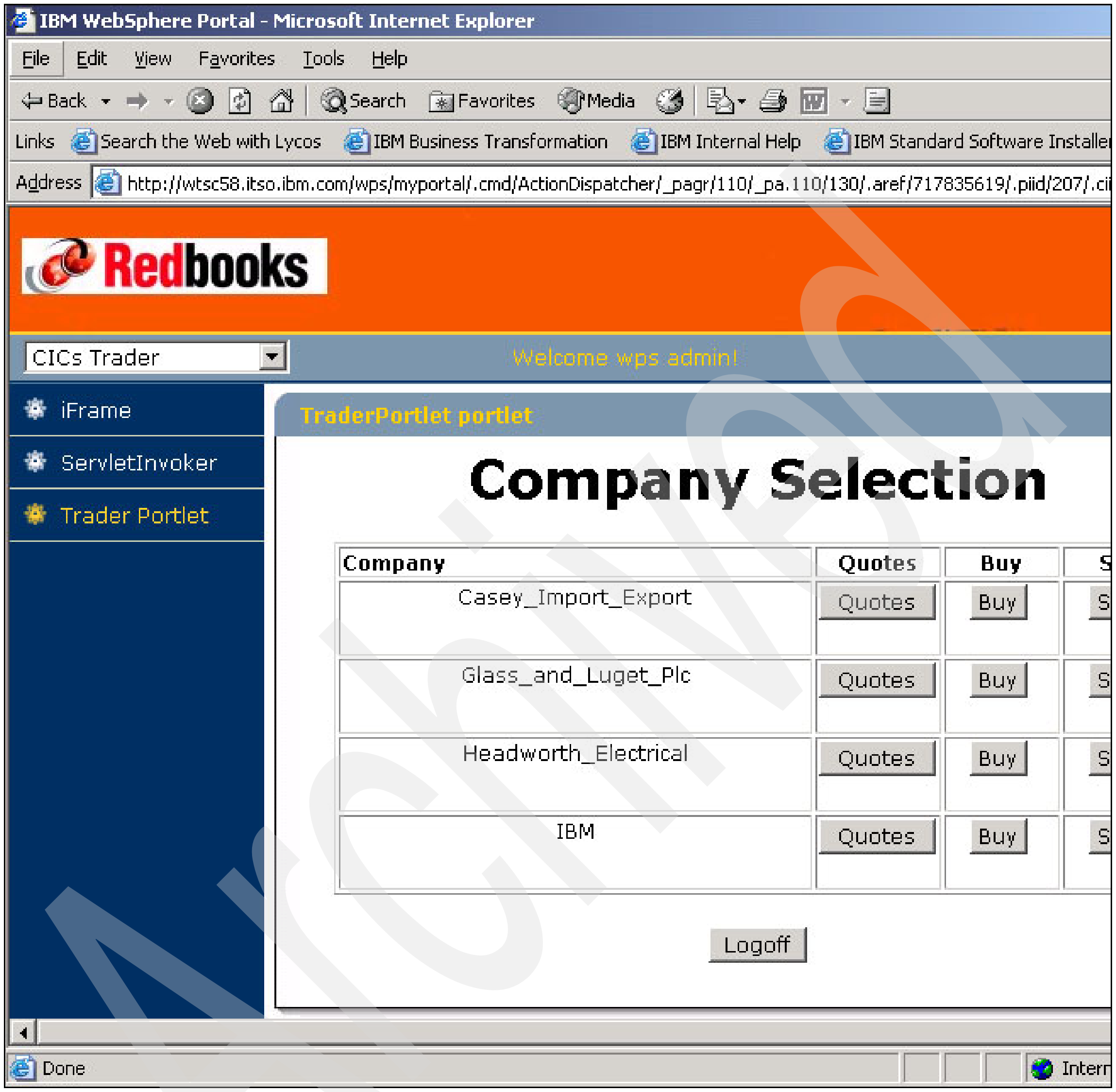
EJB location parameters :

JndiName: /PLEX58/TAMAS/Trader/TraderEJB/Trader/TraderH
NameService: com.ibm.websphere.naming.WsnInitialContextFac
ProviderURL: iiop://localhost:900/

Figure 6-63 TraderPortlet inside the Portal

As we have seen in Figure 6-36 on page 226 the EJB location parameters specified by the JNDI name are specified globally and are the same as the one we used for the original Trader Web application.

After signing on to the Trader application the Company Selection screen is received as shown in Figure 6-64 on page 280.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installer
Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispatcher/_pagr/110/_pa.110/130/.aref/717835619/.piid/207/.ci
Redbooks
CICs Trader
Welcome wps admin!
iFrame
ServletInvoker
Trader Portlet
TraderPortlet portlet
Done
Intern

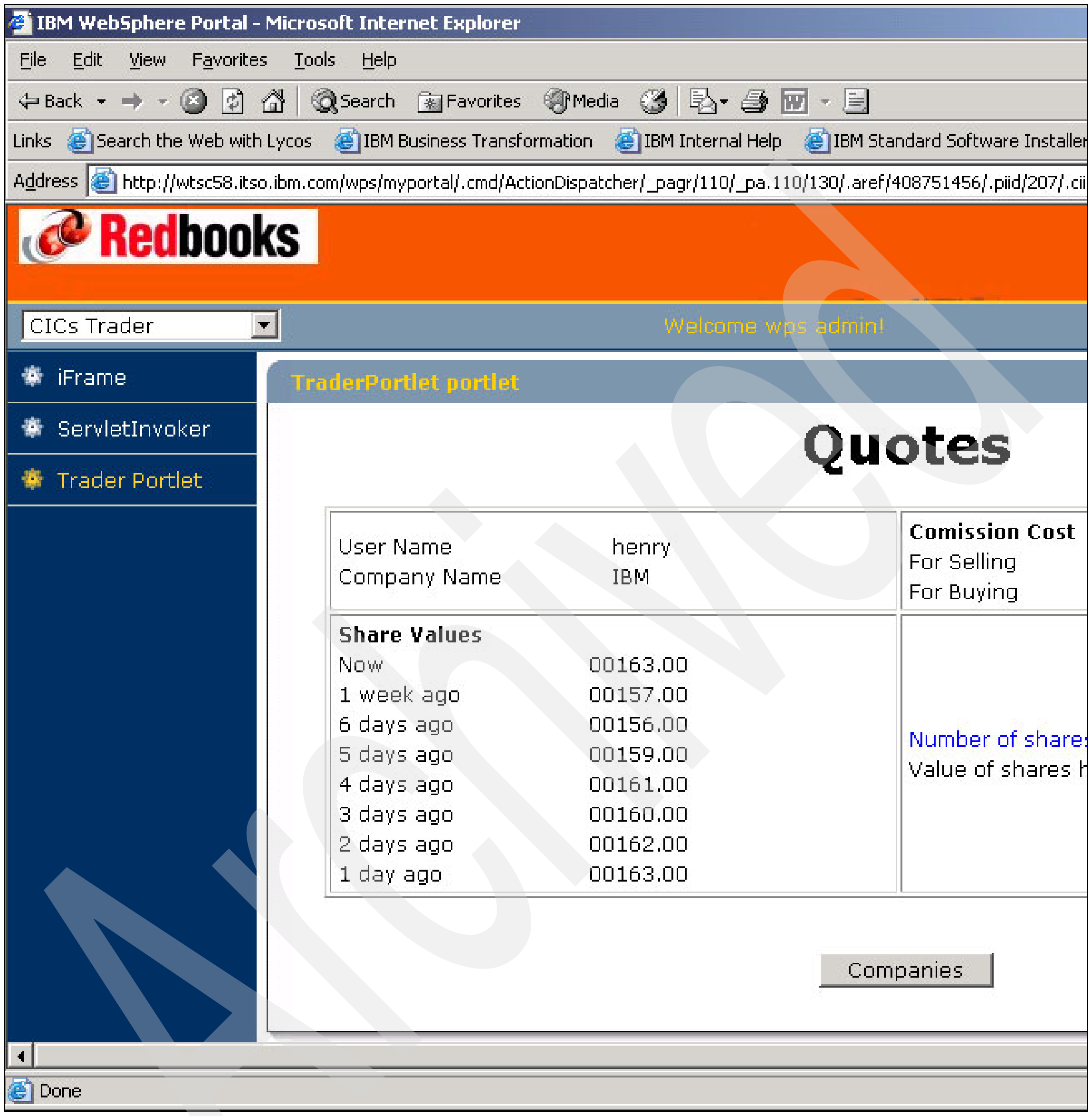
Company Selection

Company	Quotes	Buy	S
Casey_Import_Export	Quotes	Buy	S
Glass_and_Luget_Plc	Quotes	Buy	S
Headworth_Electrical	Quotes	Buy	S
IBM	Quotes	Buy	S

Logoff

Figure 6-64 Company Selection screen from the Trader Portlet

To test and exercise the portlet we chose to buy and sell some shares, first by obtaining some quotes from IBM as seen in Figure 6-65 on page 281.


IBM WebSphere Portal - Microsoft Internet Explorer

Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispatcher/_pagr/110/_pa.110/130/.aref/408751456/.piid/207/.cii

CICs Trader

Welcome wps admin!

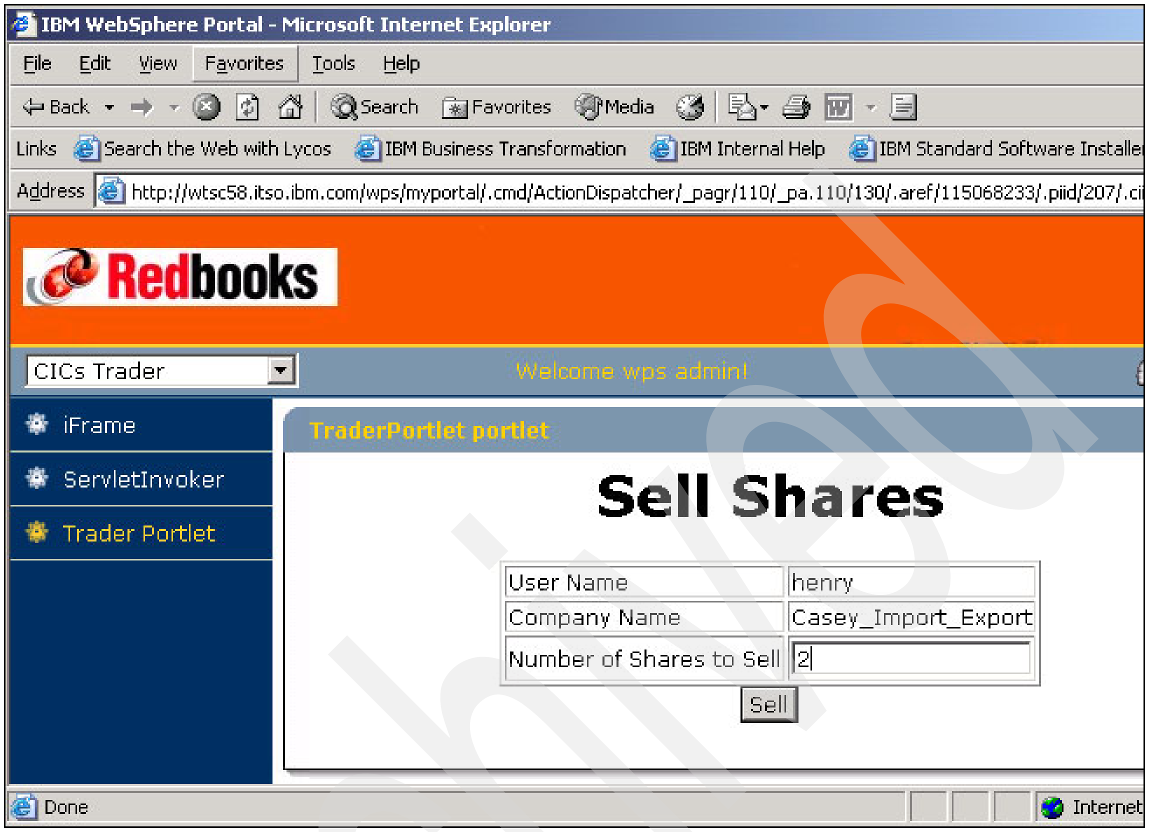
TraderPortlet portlet

Quotes

User Name	henry	**Comission Cost**
Company Name	IBM	For Selling
		For Buying
Share Values		
Now	00163.00	
1 week ago	00157.00	
6 days ago	00156.00	
5 days ago	00159.00	Number of share
4 days ago	00161.00	Value of shares h
3 days ago	00160.00	
2 days ago	00162.00	
1 day ago	00163.00	

Figure 6-65 Quotes screen from the Trader Portlet

Finally the “sell shares” portlet view is shown from the portlet in Figure 6-66 on page 282.


IBM WebSphere Portal - Microsoft Internet Explorer

Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispatcher/_pagr/110/_pa.110/130/.aref/115068233/.piid/207/.ci

Welcome wps admin!

TraderPortlet portlet

Sell Shares

User Name	henry
Company Name	Casey_Import_Export
Number of Shares to Sell	2

Figure 6-66 Sell Shares screen from the Trader Portlet

That concluded our exercise in portalizing an existing Web application to run as a portlet on WebSphere Portal. Although this was quite a simple example to work with, it demonstrated several things:

1. The logic in the servlet being converted can remain largely intact.
2. No deployment changes are needed at all to an existing J2EE application consisting of Web applications and EJBs.
3. Most of the work was in handling the portlet API for the servlet/portlet/JSPs, as might be expected. For this it helps if the developer has a working knowledge of servlet/JSP programming and understands the indirection added by the portlet API.
4. The Model-View-Controller design pattern is applicable to portlet programming, where JSPs are used for the View, portlets used for the Controller and Beans/EJBs used for the Model.

7

Portal security

This chapter discusses security aspects for WebSphere Portal and how those relate to the z/OS and OS/390 platform, and also how to use the existing RACF userids with WebSphere Portal. Note: Chapter 4., “Portal administration” on page 121 provides information about how the portal administrator can create users and groups and assign access rights to view, edit, or manage portal places, portal pages, and portlets.

7.1 WebSphere Portal security

WebSphere Portal Server on the distributed platforms has the option to use operating system (OS) level security or a Directory Server, that is, LDAP server for user authentication. The LDAP Directory Server from IBM uses DB2 database as its repository. The Portal server, LDAP server, and the database server can be co-located on the same physical machine or installed on three separate machines.

Security services for the z/OS and OS/390 platforms are usually provided by RACF or ACF2. However, WebSphere Portal server on z/OS configures the J2EE server used by Portal with a Custom User Registry (CUR) to use LDAP with a DB2 repository. Refer to *LDAP Server Administration and Usage manual,* SC24-5923, for further information on setting up LDAP on z/OS and OS/390.

Tip: The recommendation is to setup another LDAP instance on the same system where WebSphere Portal server is configured.

7.2 Portal security architecture

Security for Portal Server on z/OS is provided through the Custom User Registry (CUR) feature of WebSphere Application Server on z/OS.

The CUR feature for WebSphere Portal on the distributed platforms refers to an authentication system that does not use any of those provided with the product, for example, LDAP. On the z/OS and OS/390 platform, CUR is provided with WebSphere Portal so that it performs authentication using LDAP and not via RACF.

Portal users can typically be a few hundred existing z/OS intranet users who probably have userids managed through RACF, and potentially thousands of new internet or extranet users. The challenge is to provide these new users direct and secure access to transactions and data via the portal. These new portal users do not need RACF logon to the z/OS system, but do need to be authenticated and granted access to the specific application they need to run which is handled by WebSphere Portal via a portlet. This is where a separate registry using the Lightweight Directory Access Protocol (LDAP) as a Custom User Registry comes into play.

Figure 6-1 on page 225 shows an overview of the security implementation of WebSphere Portal Server on z/OS and OS/390.

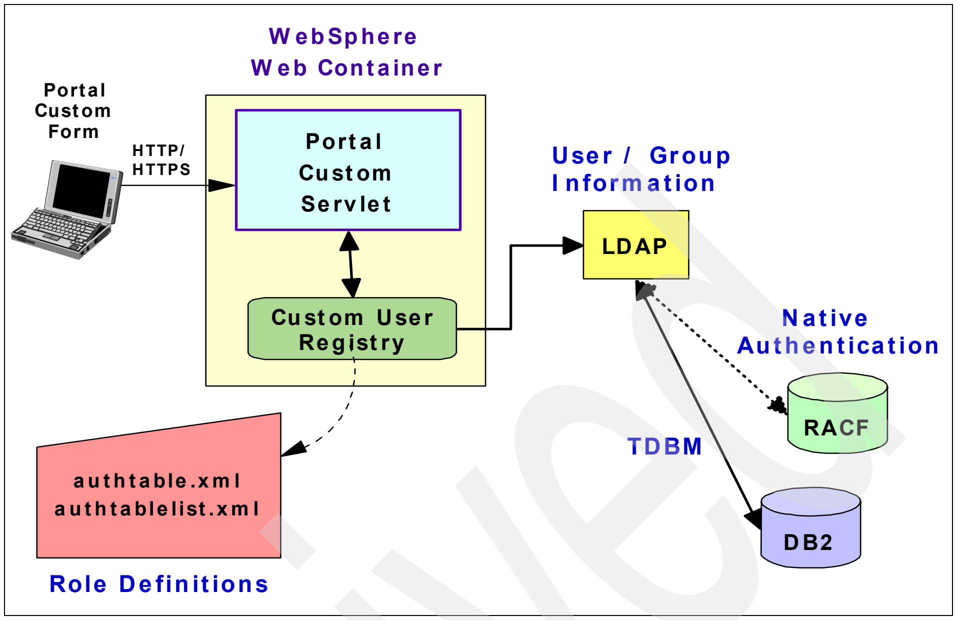


Figure 7-1 WebSphere Portal security for z/OS

7.3 Custom User Registry

WebSphere Portal server on z/OS and OS/390 supports a Custom User Registry (CUR) using the OS/390 Lightweight Directory Access Protocol server, which is part of the SecureWay® Security Server for z/OS and OS/390. This is not a z/OS System Authorization Facility (SAF) based user registry and so does not use RACF. Portal server CUR provides an implementation of the `com.ibm.websphere.security.CustomRegistry` interface which is the same interface used by WebSphere Application Server V4.01 for CUR. This allows the Portal server to service requests from users or Web clients by authenticating a user ID and password against the Custom User Registry using the Portal Custom Servlet. This is how Portal server authenticates users and provides the first layer of protection to internal portal resources, such as portlets, places and pages.

Most of the CUR configuration for the installation is done by the `wpsPost` job or the shell script `wpsPost.root.sh` in Step 9.8 of the Portal installation. During that step it creates files, such as `jvm.properties`, `webcontainer.conf` and `trace.dat` in the directory:
`WebSphere390/CB390/controlinfo/envfile/SYSPLEXNAME/WPS_SRVINST_NAME`

For CUR, an authorization table is provided via an XML file. You can find the following XML files under directory: `/usr/lpp/PortalServer/PortalServer/libapp/config`

- authtable.xml

  This XML file contains the authorizations for each Web application installed on the J2EE server for which the custom user registry is being used to authenticate requests. The authorizations are based on roleName and groupName definitions.

- authtablelist.xml
  This CUR authorization table XML file is used to define the applications and its authorization lists

The authorization table is managed by the administrator to grant users and groups access to the J2EE resources on a per application/portlet basis.

Important: You *cannot* use EJBROLE profiles defined in a SAF environment if users and groups are defined in a non-SAF based registry.

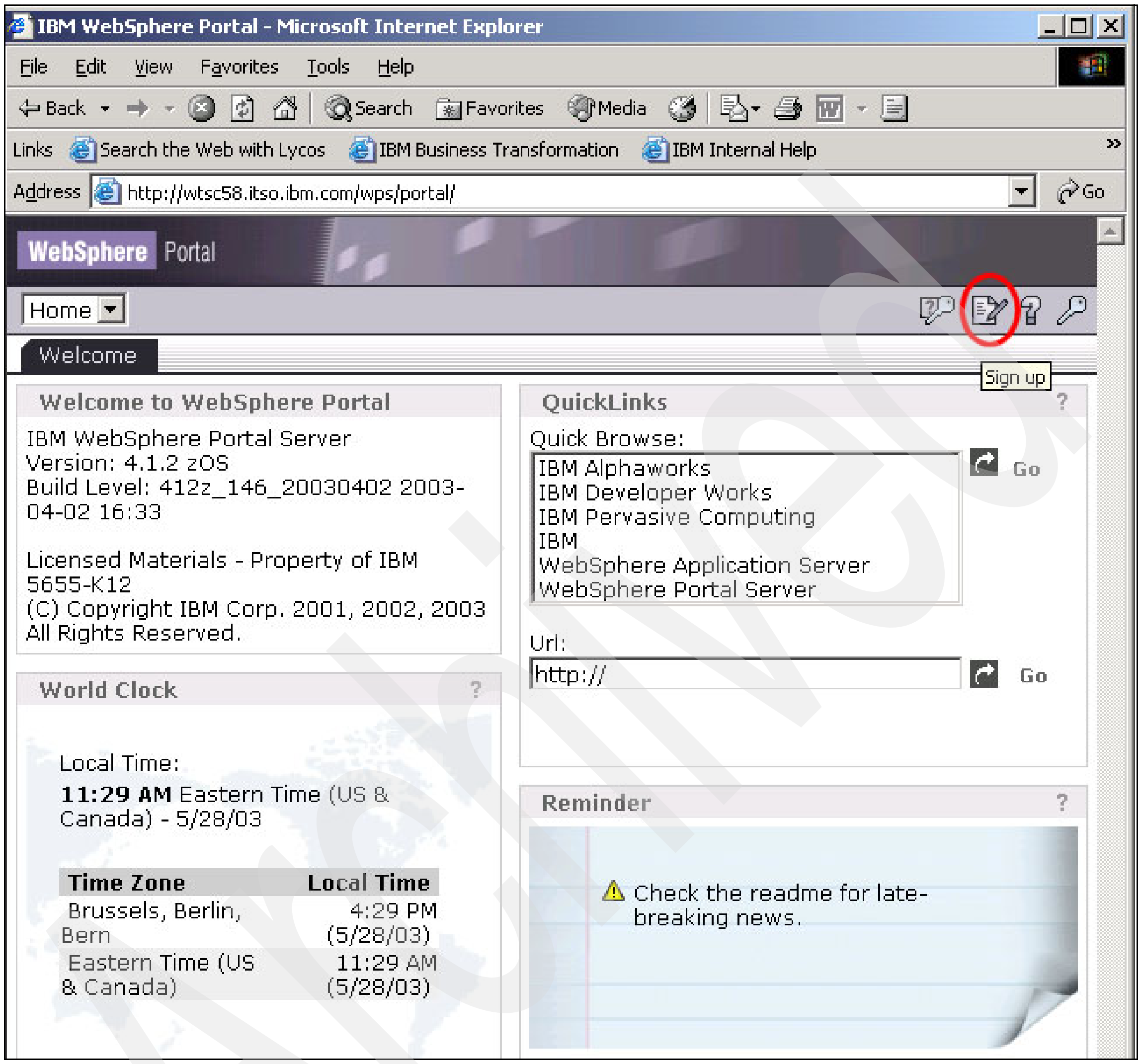
To browse or to edit these files you need to FTP the files to a workstation and then FTP them back, if any changes have been made to these files. The same applies for `puma.properties` and `um.properties` files.

7.4 How to manage user registrations

Here we describe the different options available for registering users under WebSphere Portal.

7.4.1 User self-registration

Portal users can do self-registration from the portal Web page by clicking on the 'Sign up' icon. The userid information entered on the screen will be saved as a user profile in LDAP and the DB2 TDBM backend, in the same way as if it were administered by the Portal administrator. From the Portal Web page look for the 'Sign up' icon on the navigation bar, shown as a circled icon in Figure 7-2 on page 287.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://wtsc58.itso.ibm.com/wps/portal/
Go
WebSphere Portal
Home
Welcome
Sign up

Welcome to WebSphere Portal

IBM WebSphere Portal Server
Version: 4.1.2 zOS
Build Level: 412z_146_20030402 2003-04-02 16:33

Licensed Materials - Property of IBM
5655-K12
(C) Copyright IBM Corp. 2001, 2002, 2003
All Rights Reserved.

World Clock

Local Time:
11:29 AM Eastern Time (US & Canada) - 5/28/03

Time Zone	Local Time
Brussels, Berlin, Bern	4:29 PM (5/28/03)
Eastern Time (US & Canada)	11:29 AM (5/28/03)

QuickLinks

Quick Browse:
IBM Alphaworks
IBM Developer Works
IBM Pervasive Computing
IBM
WebSphere Application Server
WebSphere Portal Server
Go

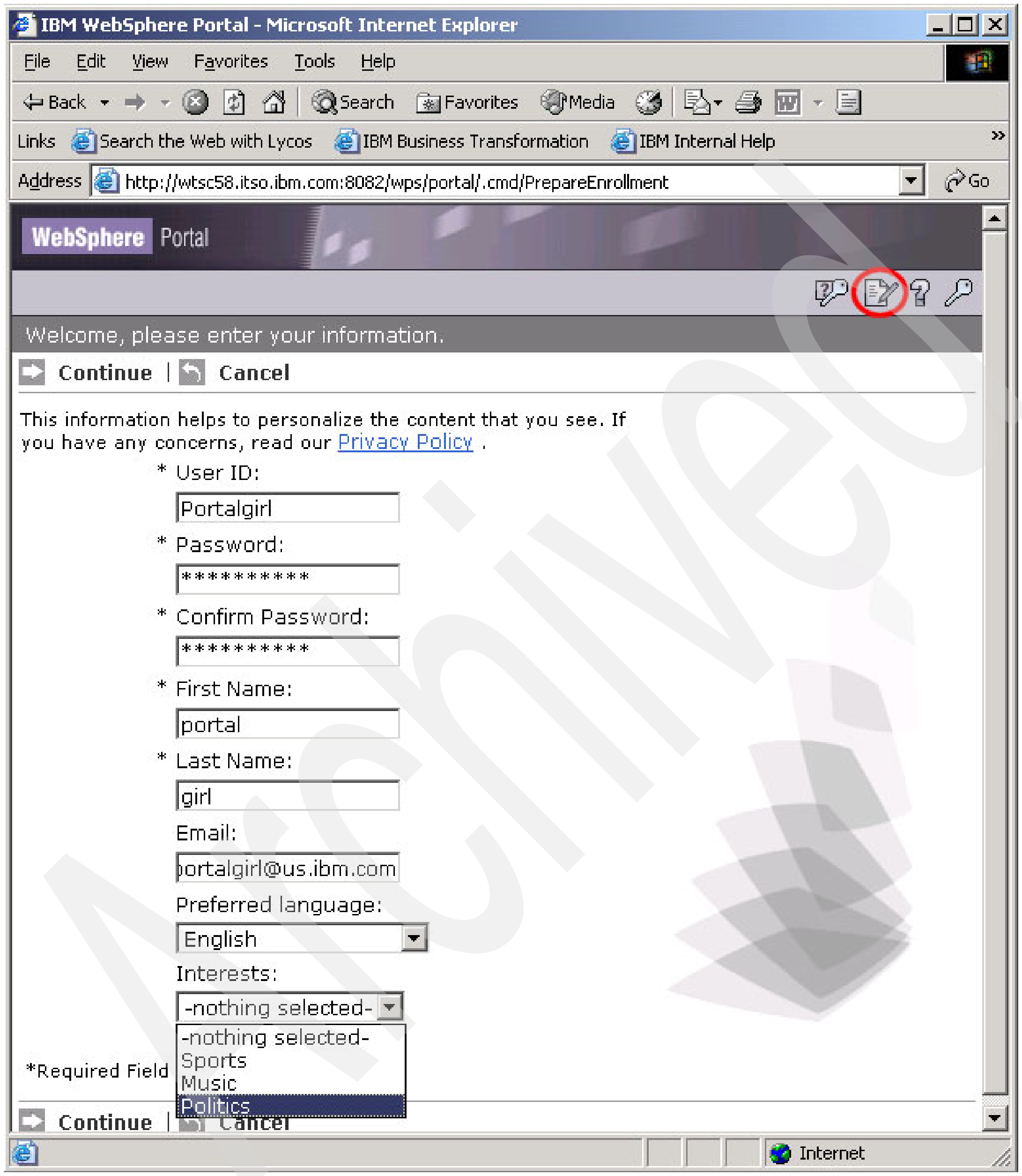
Url:
http://
Go

Reminder

Check the readme for late-breaking news.

Figure 7-2 WebSphere Portal server sign up screen

Clicking on the sign up icon takes you to the user self-registration page shown in Figure 7-3 on page 288.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://wtsc58.itso.ibm.com:8082/wps/portal/.cmd/PrepareEnrollment
Go
WebSphere Portal

Welcome, please enter your information.

Continue | Cancel

This information helps to personalize the content that you see. If you have any concerns, read our Privacy Policy .

* User ID:	Portalgirl
* Password:	**********
* Confirm Password:	**********
* First Name:	portal
* Last Name:	girl
Email:	portalgirl@us.ibm.com
Preferred language:	English
Interests:	-nothing selected-

-nothing selected-
Sports
Music
Politics

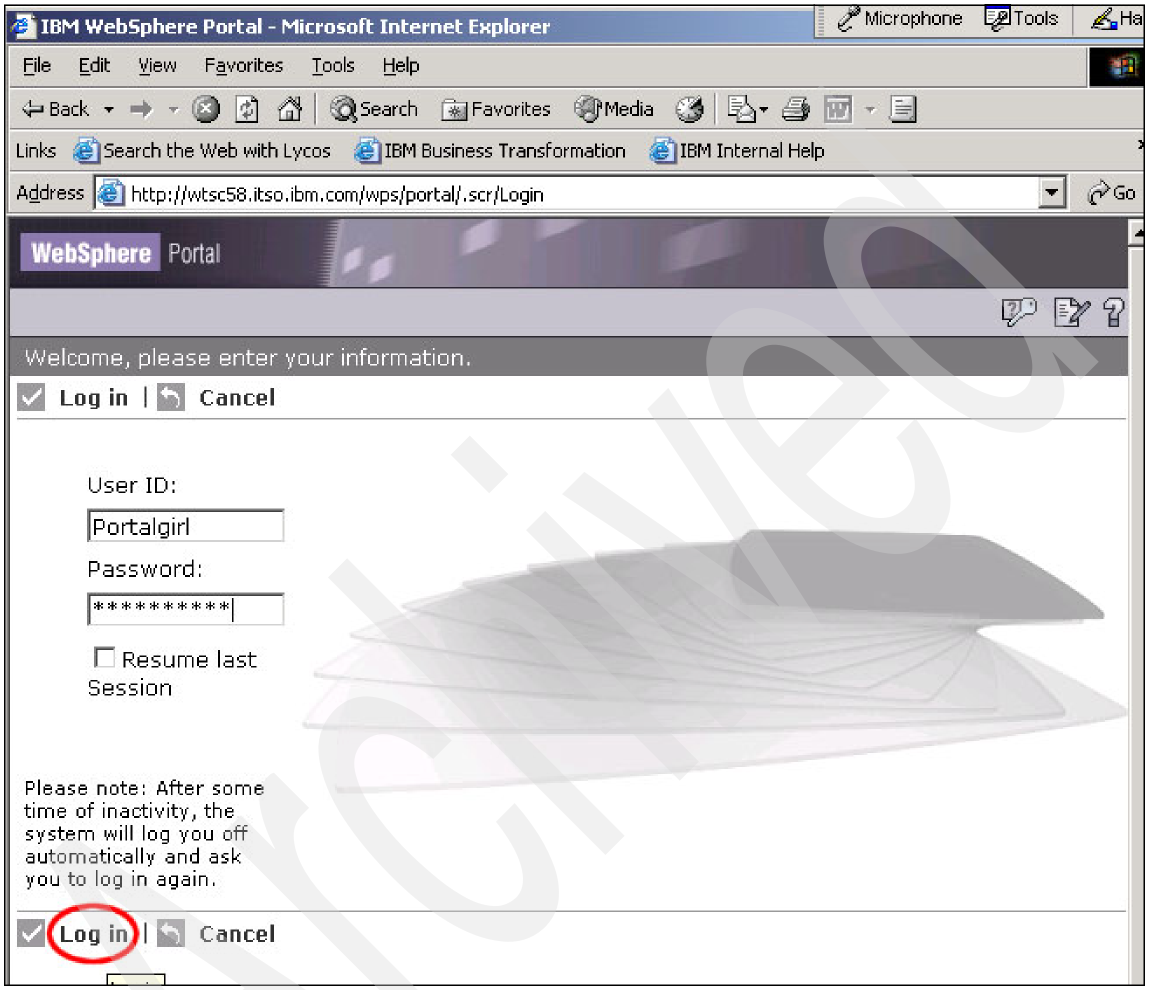
*Required Field

Continue | Cancel

Figure 7-3 WebSphere Portal Server user self-registration

After a user has provided the sign up information, click **Continue** to confirm and complete registration. Once you receive confirmation of user registration completion, you can then try to log in to the Portal Web using the login icon.

After clicking on the ‘Log in’ icon on the navigation bar, the login page shown in Figure 7-4 is used to enter user login information and login to the Portal server.


IBM WebSphere Portal - Microsoft Internet Explorer
Microphone
Tools
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Links
Search the Web with Lycos
IBM Business Transformation
IBM Internal Help
Address
Go
WebSphere Portal
Log in
Cancel
Resume last Session
Log in
Cancel

http://wtsc58.itso.ibm.com/wps/portal/.scr/Login

Welcome, please enter your information.

User ID:

Portalgirl

Password:

****

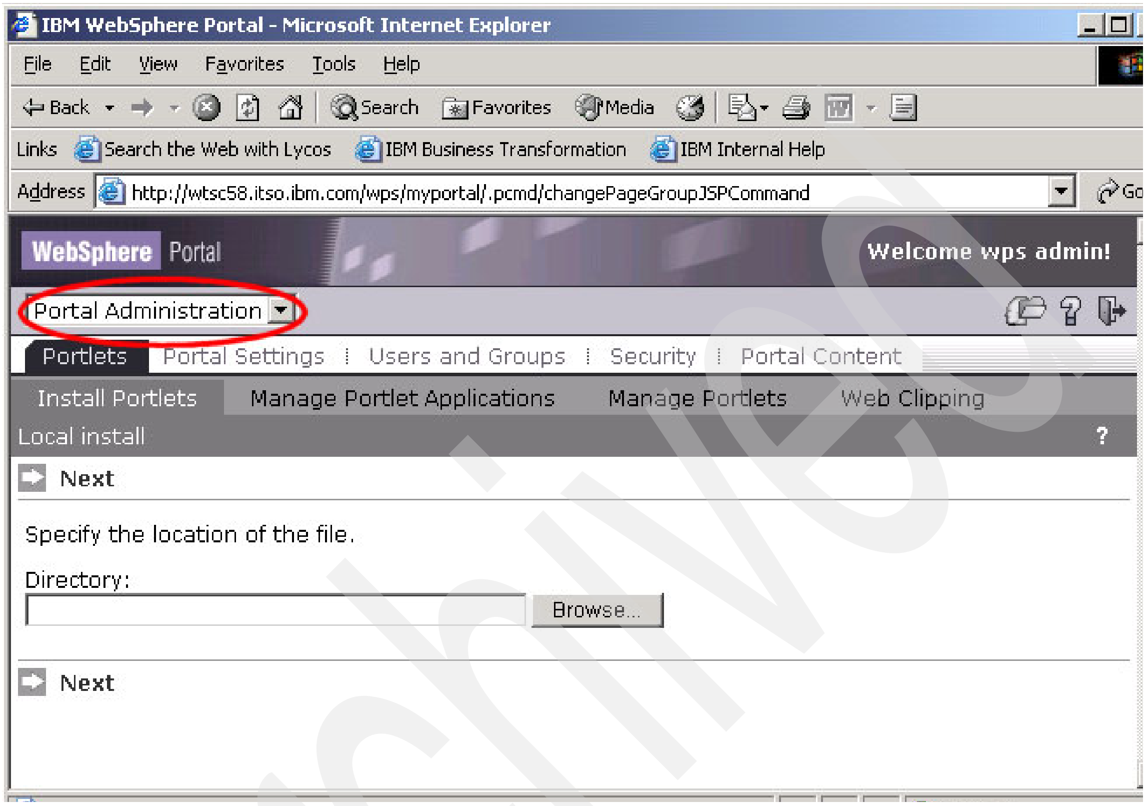
Please note: After some time of inactivity, the system will log you off automatically and ask you to log in again.

Figure 7-4 WebSphere Portal Server user login panel

Once the user is logged in to the Portal, they also have the authority to edit their own user profile.

7.4.2 Portal administrator defined registration

The portal administrator can register portal users using the system administration portlets. Also see 4.1, “Administrative privileges” on page 122, Figure 4-2 on page 124 and Figure 4-3 on page 125. After logging in as a user with administrative privileges, for example user `wpsadmin`, the Portal Administration place as shown in Figure 7-5 can be selected.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://wtsc58.itso.ibm.com/wps/myportal/.pcmd/changePageGroupJSPCommand
Go
WebSphere Portal
Welcome wps admin!
Portal Administration
Portlets Portal Settings Users and Groups Security Portal Content
Install Portlets Manage Portlet Applications Manage Portlets Web Clipping
Local install
?
Next
Browse...
Next

Specify the location of the file.

Directory:

Figure 7-5 Portal administration Portlet Panel

From the Portal Administration place, `Click` on the `Users and Groups` tab to go the page seen in Figure 7-6 on page 291 to register users.

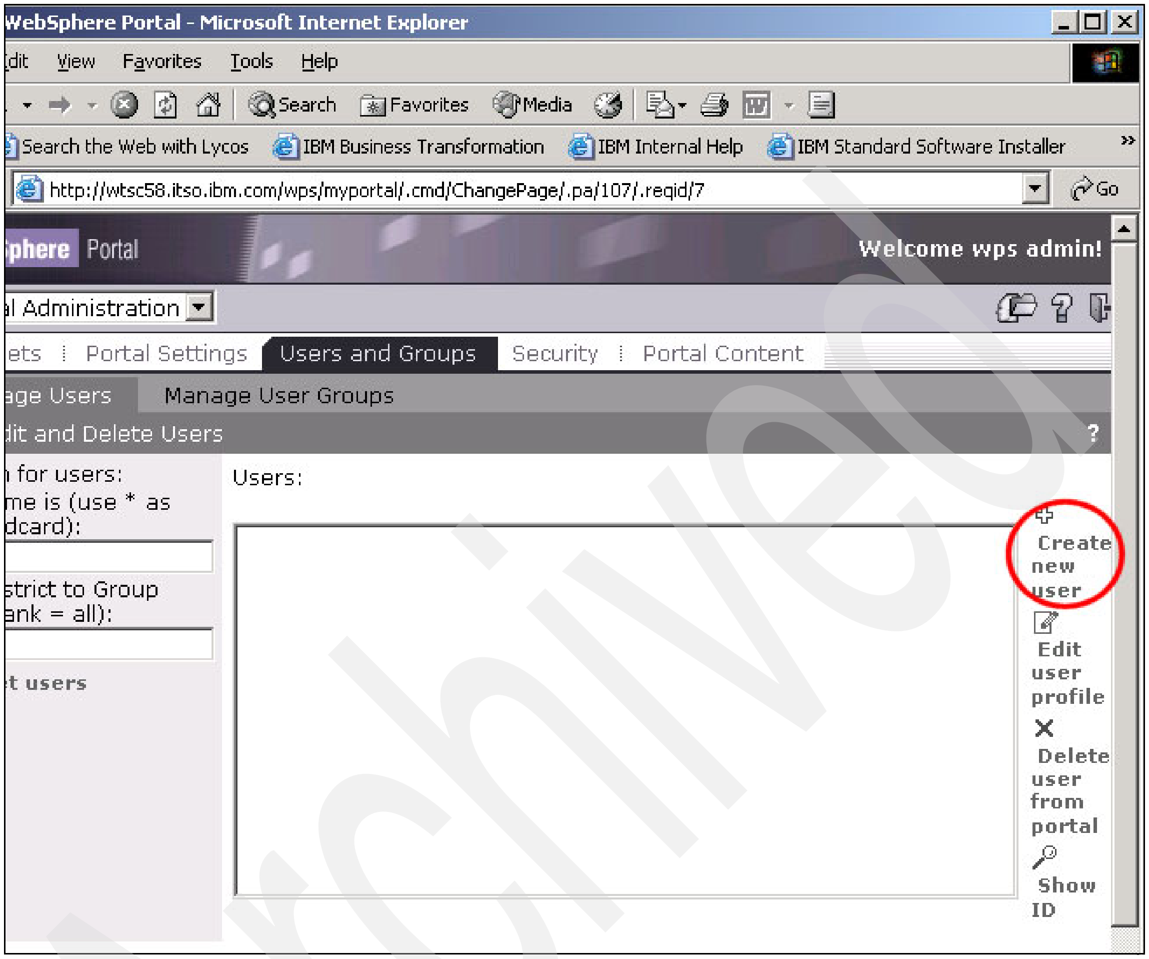
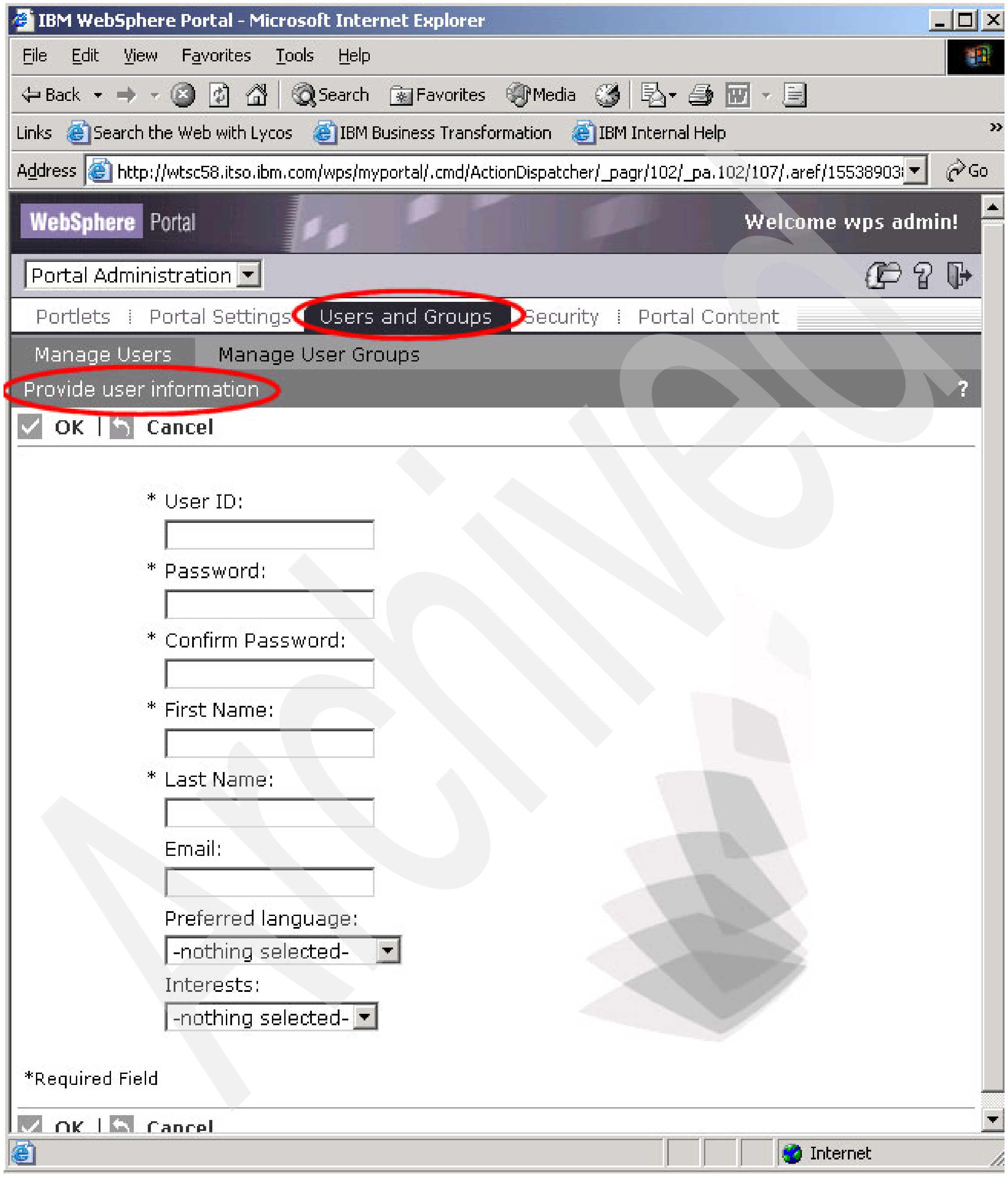


Figure 7-6 Portal administration 'User and Groups' for user registration

By clicking on `Create new user` you will be prompted with the self-registration screen as shown in Figure 7-3 on page 288, where the administrator can create new user information.


IBM WebSphere Portal - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://wtsc58.itso.ibm.com/wps/myportal/.cmd/ActionDispatcher/_pagr/102/_pa.102/107/.aref/15538903
Go
WebSphere Portal
Welcome wps admin!
Portal Administration
Portlets Portal Settings Users and Groups Security Portal Content
Manage Users Manage User Groups
Provide user information
OK Cancel
Internet

* User ID:

* Password:

* Confirm Password:

* First Name:

* Last Name:

Email:

Preferred language: -nothing selected-

Interests: -nothing selected-

*Required Field

Figure 7-7 Add user and Provide User Information

7.4.3 IBM LDAP Data Management Tool (DMT)

The IBM SecureWay Directory Server has an LDAP client tool called Directory Management Tool (DMT). The DMT is shipped on CD-ROMs as a part of WebSphere Portal Server for z/OS and OS/390 development environment. DMT is a Java client graphical user interface that allows an administrator to manage LDAP directories on multiple LDAP servers. DMT supports the following functions:

- Displaying server properties and rebinding to the server
- Listing, adding, editing and deleting schema attributes and object classes
- Listing, adding, editing and deleting directory entries, for example users
- Modifying directory entry ACLs
- Searching the directory tree

Adding, modifying and deleting users in the LDAP is also supported from the Portal server using the Portal Administration portlets.

The DMT can be installed from CD-ROMs 20 and 21 to the distributed platform of choice. To install only the DMT client, select the custom install option, and de-select options to install the LDAP server and DB2 database. After starting the DMT, if the LDAP server is not local then you must click on `Add server` to connect to the directory server. You then would need to enter the server name and LDAP port number, which in our case was `wtsc58.itso.ibm.com` and `2389` respectively.

To add a new entry, for example a new user for Portal Server, first you need to connect, or rebind to the server using an ID that has LDAP administrative privilege. Then browse the directory tree, select the users attribute `cn=users` and click on the **`Add`** button as shown in Figure 7-8 on page 294.

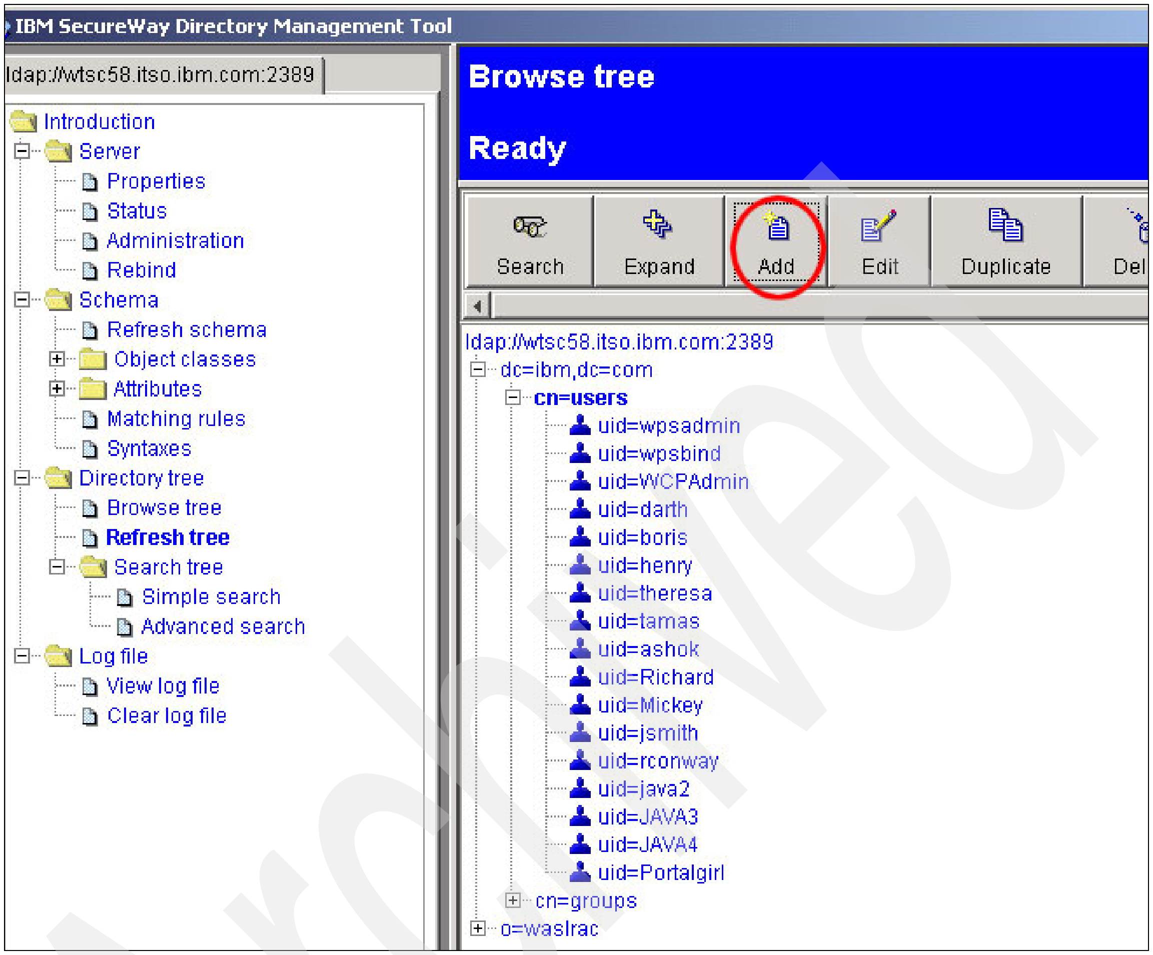
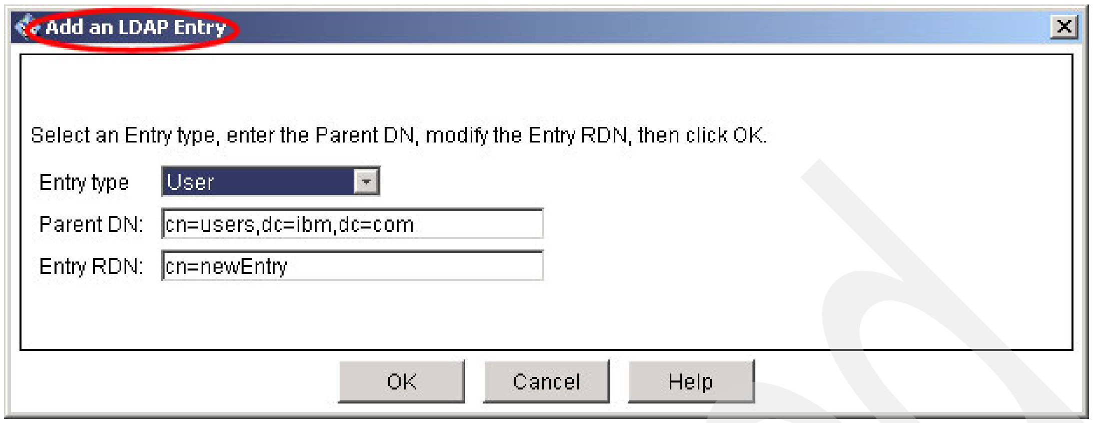


Figure 7-8 DMT Directory tree

After clicking on Add the dialog window shown in Figure 7-9 on page 295 is received.


Add an LDAP Entry

Select an Entry type, enter the Parent DN, modify the Entry RDN, then click OK.

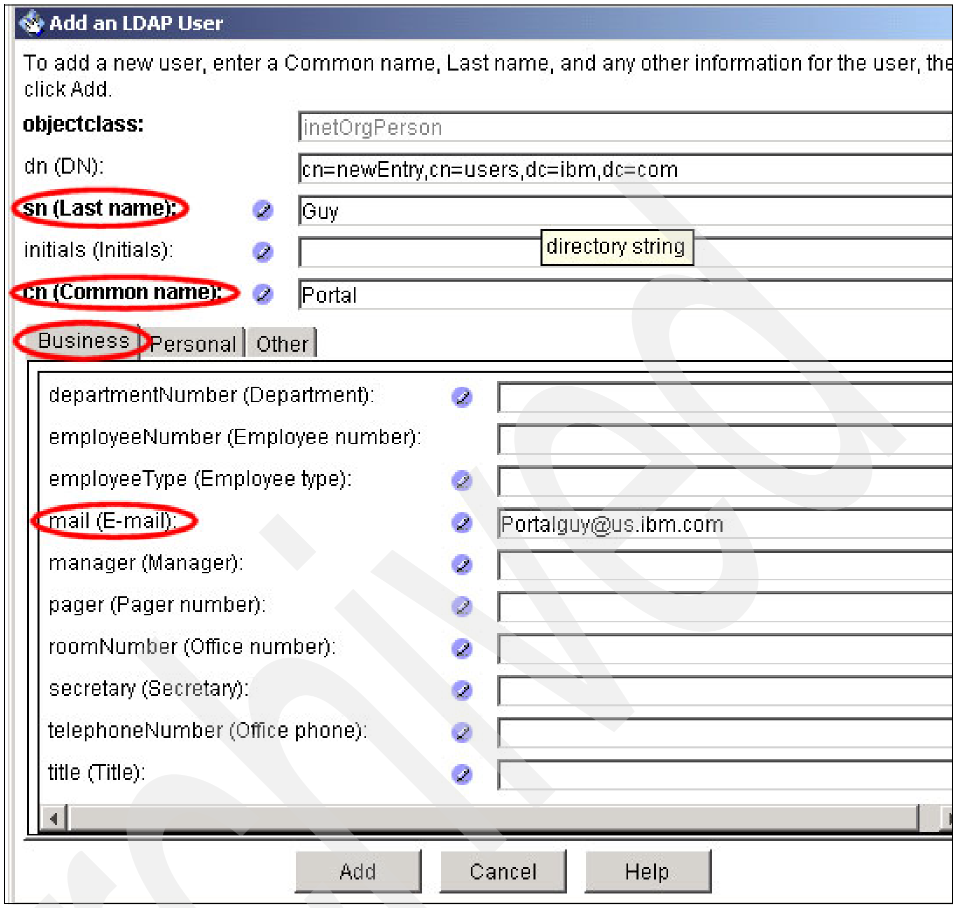
Entry type: User

Parent DN: cn=users,dc=ibm,dc=com

Entry RDN: cn=newEntry

Figure 7-9 Add an LDAP Entry - Entry type 'User'

After checking and validating the base parameters, click **OK** to continue onto the window shown in Figure 7-10 on page 296.



Add an LDAP User

To add a new user, enter a Common name, Last name, and any other information for the user, the
click Add.

objectclass:	inetOrgPerson
dn (DN):	cn=newEntry,cn=users,dc=ibm,dc=com
sn (Last name):	Guy
initials (Initials):	
cn (Common name):	Portal

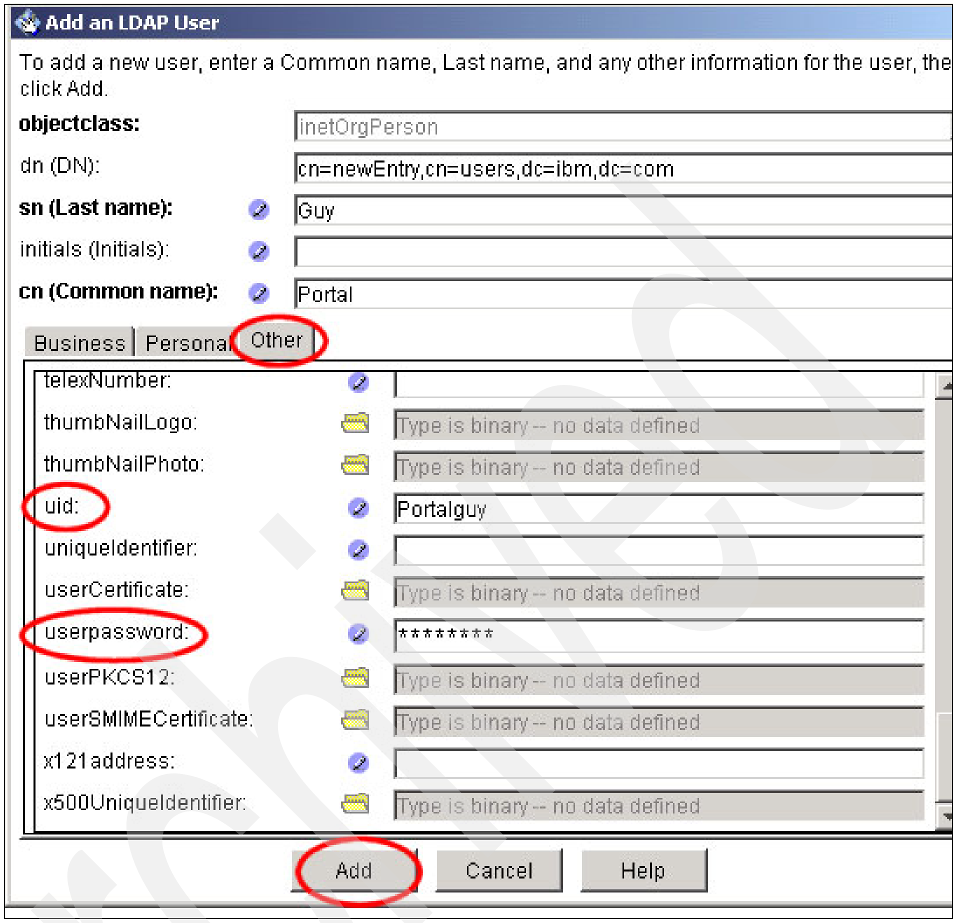
departmentNumber (Department):	
employeeNumber (Employee number):	
employeeType (Employee type):	
mail (E-mail):	Portalguy@us.ibm.com
manager (Manager):	
pager (Pager number):	
roomNumber (Office number):	
secretary (Secretary):	
telephoneNumber (Office phone):	
title (Title):	

Figure 7-10 Add an LDAP User

For Portal Server the minimum attributes that need to be completed are:

- sn: Last name
- cn Common name
- uid: userid
- userpassword: password

After providing values for these parameters click on the **Add** shown in Figure 7-11 on page 297 button to proceed.


Add an LDAP User

To add a new user, enter a Common name, Last name, and any other information for the user, the click Add.

objectclass:	inetOrgPerson
dn (DN):	cn=newEntry,cn=users,dc=ibm,dc=com
sn (Last name):	Guy
initials (Initials):	
cn (Common name):	Portal

telexNumber:	
thumbNailLogo:	Type is binary -- no data defined
thumbNailPhoto:	Type is binary -- no data defined
uid:	Portalguy
uniqueIdentifier:	
userCertificate:	Type is binary -- no data defined
userpassword:	********
userPKCS12:	Type is binary -- no data defined
userSMIMECertificate:	Type is binary -- no data defined
x121address:	
x500UniqueIdentifier:	Type is binary -- no data defined

Figure 7-11 Adding a LDAP user

Other attributes, for example, `email` and `interests` as shown in Figure 7-3 on page 288 for self registration, need not be added to LDAP. This is because the CUR implementation performed by Portal Server only uses LDAP for authentication where it is checking for the existence of a user and the correct password. Anything other than userid and password information are part of the user profile which is stored in the Portal database and not LDAP.

Note: The IBM SecureWay Directory Management Tool used in this redbook project, and provided on the CD-ROM is part of IBM Directory Server V4.1.

For customers that already have their corporate users in an existing LDAP registry on z/OS, for example Intranet users,.you can simply point to the existing LDAP registry and make sure the LDAP is configured for Portal use.

7.5 z/OS LDAP Native Authentication

The native authentication feature uses LDAP with TDBM, and any of the V3 person type object classes, since they have the appropriate attributes for this method of authentication. Portal Server still uses CUR and accesses LDAP for its authentication step, but from a z/OS and OS/390 platform perspective the authentication is actually performed by RACF using all its usual stringent rules. From a management perspective there is no need for administration of multiple registries or synchronization of passwords. Native authentication is implemented through the addition of an LDAP attribute called `ibm-nativeId`.

More importantly, from a WebSphere Portal perspective, RACF users and non-RACF users can be defined in the same LDAP directory and portal users are unaware of any differences from what is normally done to log into the portal.

Attention: The z/OS LDAP Native Authentication setup is optional. This is *not* a requirement for WebSphere Portal Server configuration and installation.

7.5.1 The scenario

If all portal users are created from new, then all those userids will be created in LDAP server as normal. More often than not, z/OS enterprises will have existing users in RACF who need access to the portal along with the newly registered external portal users. Therefore a mechanism is needed to be able to authenticate using both LDAP and RACF.

7.5.2 The proposed solution

As was mentioned in Chapter 1, “Benefits of having WebSphere Portal on z/OS” on page 16 one of the benefits of having the LDAP server running on the z/OS system is that you can do native authentication. Native authentication allows connection between the LDAP server and RACF wherein the userid and password that is used to authenticate to LDAP is actually passed to the System Security Server to be verified. This setup allows new internet or extranet portal customers to authenticate directly against the LDAP server, while existing RACF intranet users would be authenticated using their RACF userid and password. The architecture at a high level is shown Figure 7-12 on page 299, where userid `java3` is also configured as the `native-id` attribute in LDAP, resulting in the authentication request going to RACF. For a complete description of how this is done refer to *Putting the Latest z/OS Security Features to Work,* SG24-6540.

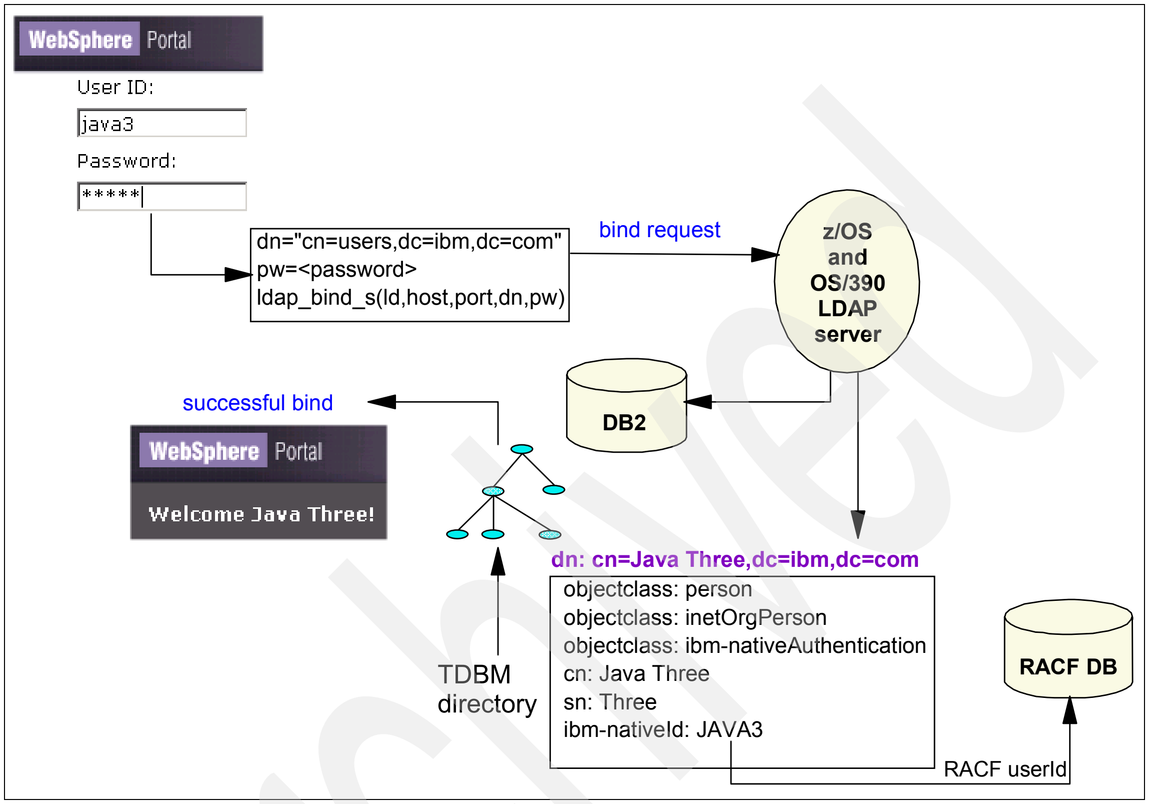


Figure 7-12 Portal login binding to LDAP server using RACF Native Authentication

7.5.3 Steps to implement the solution

The way we implemented the solution was basically to re-run some of the LDAP configuration steps described in Step seven: Configure LDAP server. There are two parts:

1. Add the object class to the LDAP server.
2. Add the attribute to user entries in LDAP.

1) Add the object class to the LDAP server

We show the steps performed on our z/OS system to add the required object class to the LDAP server.

Create a specific IBM schema

The schema to be added is an IBM schema that includes definitions for handling Native Authentication from LDAP to RACF. It is stored in an LDAP Directory Interchange Format (LDIF) file called `schema.IBM.ldif` that is found in the /tmp directory. We made a copy of that file in the /tmp directory and called it `schema.IBM.itso.ldif`. The only modifications made to it are shown in Example 7-1.

We changed the DN to match the DN that we used for WebSphere Portal on z/OS and OS/390 namely, "`dc=itso,dc=com`". By default the file contains `O=WASLRAC` because it applies to the WebSphere Application Server DN which we were not using.

Example 7-1 schema.IBM.itso.ldif file extract

```
(...)
dn: dc=ibm,dc=com
changetype:modify
add:attributetypes
(...)
```

Create a specific bbomtdbm.sh script

In order to apply this new schema you can copy and reuse the existing shell script of WebSphere Application Server and point to our new LDIF file.

Note: Do not change the original LDIF file because you will not be able to build a new LDAP server for WebSphere Application Server.

We copied `/tmp/bboldtbm.sh` to `/tmp/bbomtdbmitso.sh` and made the modifications as shown in Example 7-2:

Example 7-2 bboltdbmitso.sh file

```
LDAPUSERID=$1
LDAPPWD=$2
ldapmodify -h 127.0.0.1 -p 2389 -D $LDAPUSERID -w $LDAPPWD -f
/tmp/schema.IBM.itso.ldif
```

Important: Note that there are only three lines in Example 7-2, such that each of the lines beginning with `ldapmodify` must be contained wholly within a single line.

Modify the schema

In order to modify the schema you need to run the modified `bboltdmmitso.sh` shell script. Make sure you are logged to the system as a user with UID=0 and run the script as shown:

```
bboltdbmitso.sh "cn=CBAdmin" <password>
```

The generic syntax of that command is:

```
bboltdbmitso.sh "<LDAP_ADMIN>" <LDAP_ADMIN_PWD>
```

The output of that script should tell you that your DN was modified.

Enable Native Authentication

To enable native authentication in Portal for z/OS LDAP server you need to modify the property file (slapd.conf) of the specific LDAP server used for portal. That file is located <WAS_HOME>/<SYSPLEX_NAME>/etc/ldap and is called `<SYSPLEX_NAME>.bboslapd.conf.portal`.

In our case it was:

```
/WebSphere390/CB390/PLEX58/etc/ldap/SC58.bboslapd.conf.portal.
```

Modify the file as shown in Example 7-3 in bold. Instead of “`dc=ibm,dc=com`” you need to specify the DN suffix you chose for your z/OS portal. It has to match the suffix that is recognized by TDBM which you should find at the end of your file.

Example 7-3 Enable Native Authentication in <SYSPLEX_NAME>.bboslapd.conf.portal

```
(...)
#--------------------------------------------------
# tdbm database definitions
#--------------------------------------------------
database tdbm  GLDBTDBM
servername   DB2A
databasename BBOLDAT
dbuserid     BBOLDAT
attroverflowsize 255
extendedgroupsearching on
usenativeauth selected
nativeauthsubtree dc=ibm,dc=com
nativeupdateallowed off
suffix   "dc=ibm,dc=com"
```

Configuration options of the three parameters are as follows:

- useNativeAuth <selected | all | off>
  - `selected` - only entries located in native subtrees that contain the ibm-nativeId attribute will be subject to native authentication

  - `all` - every entry in native subtrees will use native authentication and the RACF ID can be specified with either the ibm-nativeId or UID attribute
  - `off` - Native Authentication is disabled
- nativeAuthSubtree <all | DN>
  - `all` - the entire TDBM directory will use native authentication
  - `DN` - subtree containing entries that will use native authentication. This option might be specified multiple times to allow for several native authenticating sub-trees.
- nativeUpdateAllowed <on / yes | off / no>
  - `on / yes` - allow updating of the native RACF password
  - `off / no` - not allowed to update the native RACF password

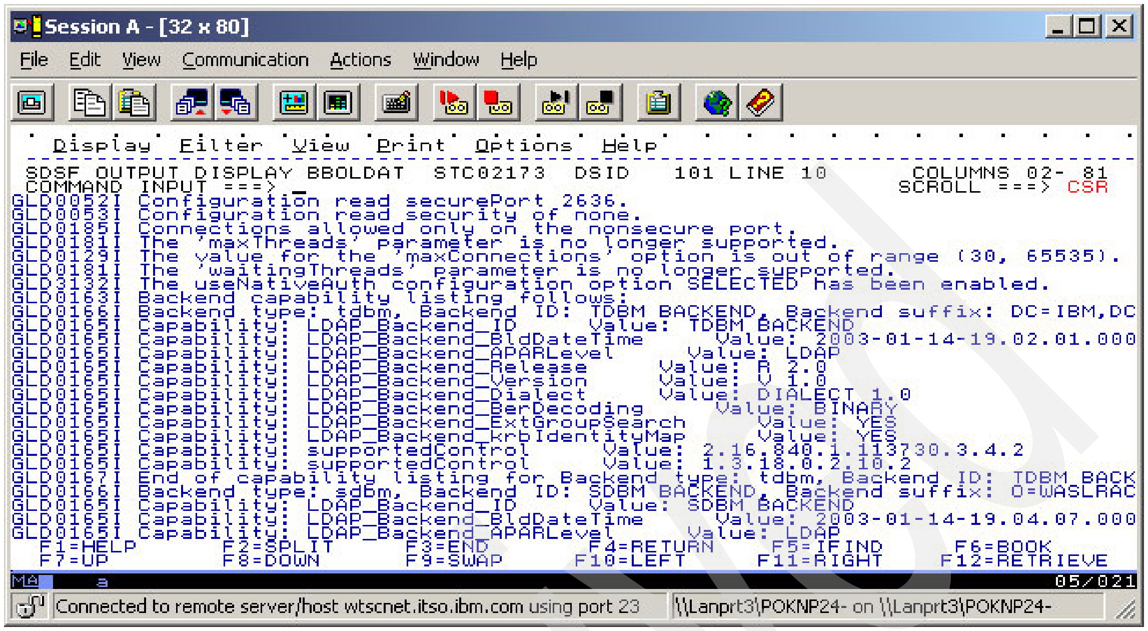
Note: Our recommendation is to not allow native RACF password updates.

Restart Portal for z/OS LDAP server

The LDAP server for WebSphere Portal on z/OS and OS/390 should be restarted, for the changes that were made to enable Native authentication, to take effect.

- First stop the LDAP server by issuing the following command: **`/P BBOLDAT`**
- Then restart the LDAP server by issuing the start command: **`/S BBOLDAT`**.

Examine the log for the BBOLDAT job. It should show that Native Authentication is enabled as shown in Example 7-13 on page 303. Notice the line that reads - `GLD3132I The useNativeAuth configuration option SELECTED has been enabled`.


Session A - [32 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help

```
 SDSF OUTPUT DISPLAY BBOLDAT  STC02173  DSID   101 LINE 10      COLUMNS 02- 81
 COMMAND INPUT ===> _                                          SCROLL ===> CSR
GLD0052I Configuration read securePort 2636.
GLD0053I Configuration read security of none.
GLD0185I Connections allowed only on the nonsecure port.
GLD0181I The 'maxThreads' parameter is no longer supported.
GLD0129I The value for the 'maxConnections' option is out of range (30, 65535).
GLD0181I The 'waitingThreads' parameter is no longer supported.
GLD3132I The useNativeAuth configuration option SELECTED has been enabled.
GLD0163I Backend capability listing follows:
GLD0166I Backend type: tdbm, Backend ID: TDBM BACKEND, Backend suffix: DC=IBM,DC
GLD0165I Capability: LDAP_Backend_ID       Value: TDBM BACKEND
GLD0165I Capability: LDAP_Backend_BldDateTime     Value: 2003-01-14-19.02.01.000
GLD0165I Capability: LDAP_Backend_APARLevel     Value: LDAP
GLD0165I Capability: LDAP_Backend_Release     Value: R 2.0
GLD0165I Capability: LDAP_Backend_Version     Value: V 1.0
GLD0165I Capability: LDAP_Backend_Dialect     Value: DIALECT 1.0
GLD0165I Capability: LDAP_Backend_BerDecoding     Value: BINARY
GLD0165I Capability: LDAP_Backend_ExtGroupSearch     Value: YES
GLD0165I Capability: LDAP_Backend_krbIdentityMap     Value: YES
GLD0165I Capability: supportedControl      Value: 2.16.840.1.113730.3.4.2
GLD0165I Capability: supportedControl      Value: 1.3.18.0.2.10.2
GLD0167I End of capability listing for Backend type: tdbm, Backend ID: TDBM BACK
GLD0166I Backend type: sdbm, Backend ID: SDBM BACKEND, Backend suffix: O=WASLRAC
GLD0165I Capability: LDAP_Backend_ID       Value: SDBM BACKEND
GLD0165I Capability: LDAP_Backend_BldDateTime     Value: 2003-01-14-19.04.07.000
GLD0165I Capability: LDAP_Backend_APARLevel     Value: LDAP
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP    F10=LEFT     F11=RIGHT    F12=RETRIEVE
MA    a                                                                05/021
```

Connected to remote server/host wtscnet.itso.ibm.com using port 23 | \\Lanprt3\POKNP24- on \\Lanprt3\POKNP24-

Figure 7-13   Portal for z/OS LDAP server startup with Native Authentication enabled

2) Add the attribute to user entries in LDAP

Using the DMT as an example we show how to add the ibm-nativeId attribute to users in LDAP.

Modifying the user attribute in LDAP

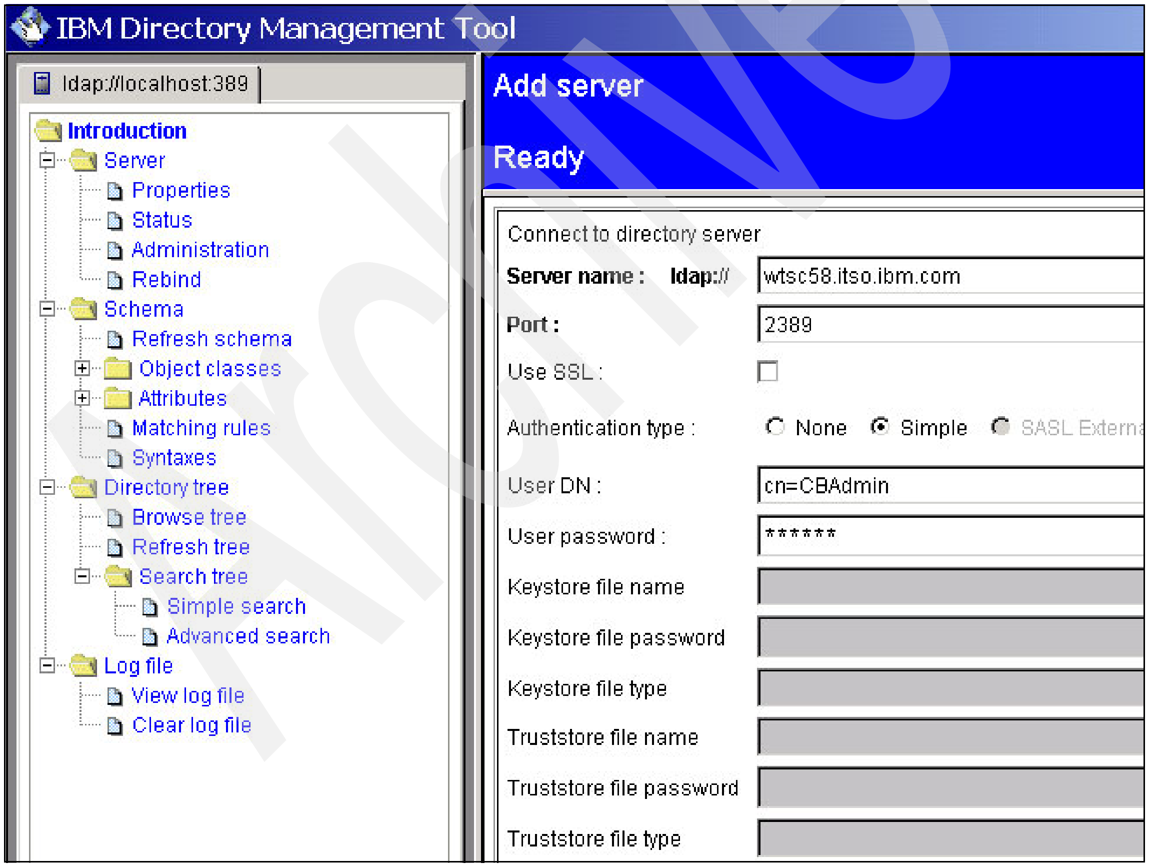
IBM Directory Server (IDS) comes with a client tool that can connect to any LDAP server and allows you to manage the directory. It is known as the Directory Management Tool (DMT) and the executable is found in `<IDS_HOME>/bin` directory on the distributed platform. We used this graphical user interface (GUI) tool to see the changes made in the previous step and to manually add the attribute to a user entry.

There are other LDAP browsers that one could use.

Note: The manual steps to add an attribute described here are simply to demonstrate what needs to be done. In a production environment we recommend using some kind of a LDAP modification script that can handle a large number of user entries in bulk.

One possibility is to generate a flat file output of RACF database using the IRRDBU00 utility. Then use a program to create a LDIF file from that IRRDBU00 output and import it into LDAP.

1. Start DMT client and connect to the LDAP server on z/OS and OS/390. We used port 2389 to connect to the LDAP server on our server.
2. Use Simple Authentication to bind to the server, by giving the User DN and User password of the LDAP administrative user, so that you will be able to view and edit the LDAP server tree. See Figure 7-14.



Connect to directory server

Server name : ldap:// wtsc58.itso.ibm.com

Port : 2389

Use SSL : ☐

Authentication type : None [✓ Simple] SASL Externa

User DN : cn=CBAdmin

User password : ******

Keystore file name

Keystore file password

Keystore file type

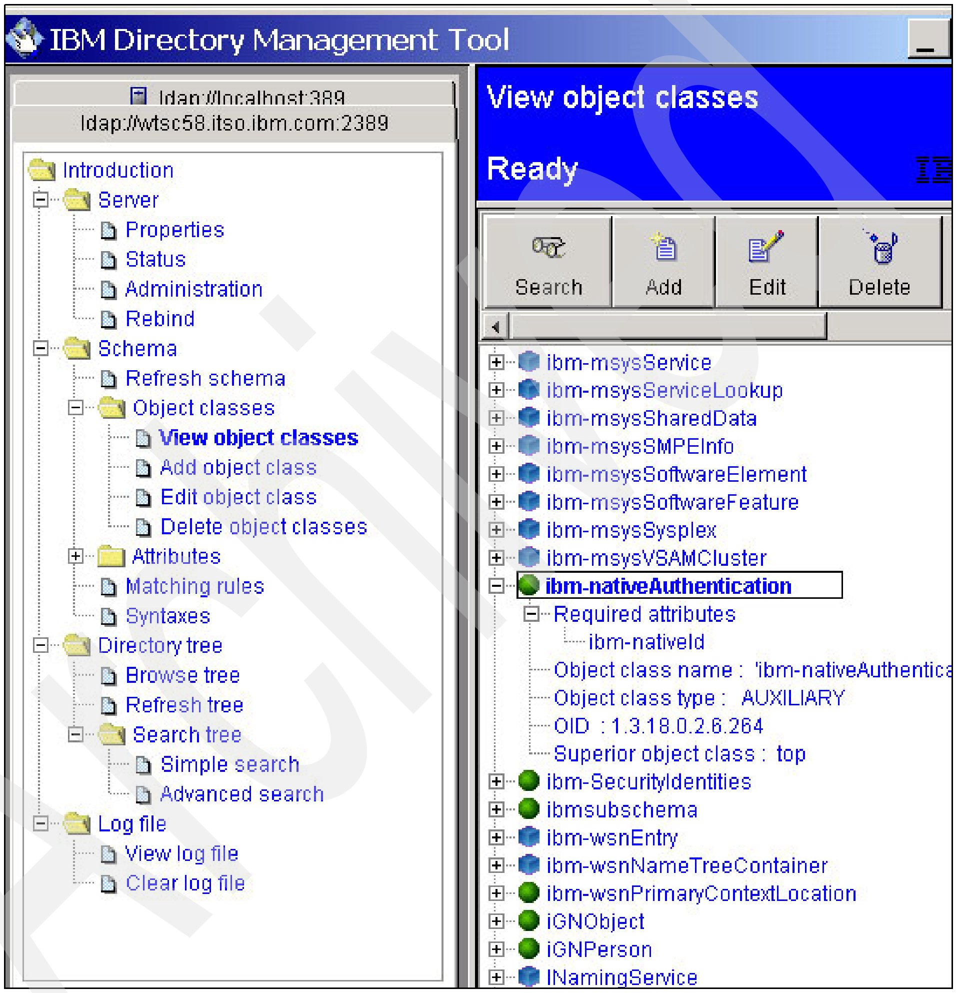
Truststore file name

Truststore file password

Truststore file type

Figure 7-14 Adding LDAP server in DMT

3. To view the object class that was added to LDAP in the previous step on z/OS, in the left navigation pane, navigate down to `Schema->Object classes->View object classes`. All the object classes will be displayed in the main window.
4. Find `ibm-nativeAuthentication` object class and click on the plus sign to see the details of the schema as shown in Figure 7-15. Notice the object class has one required attribute namely, `ibm-nativeId`.

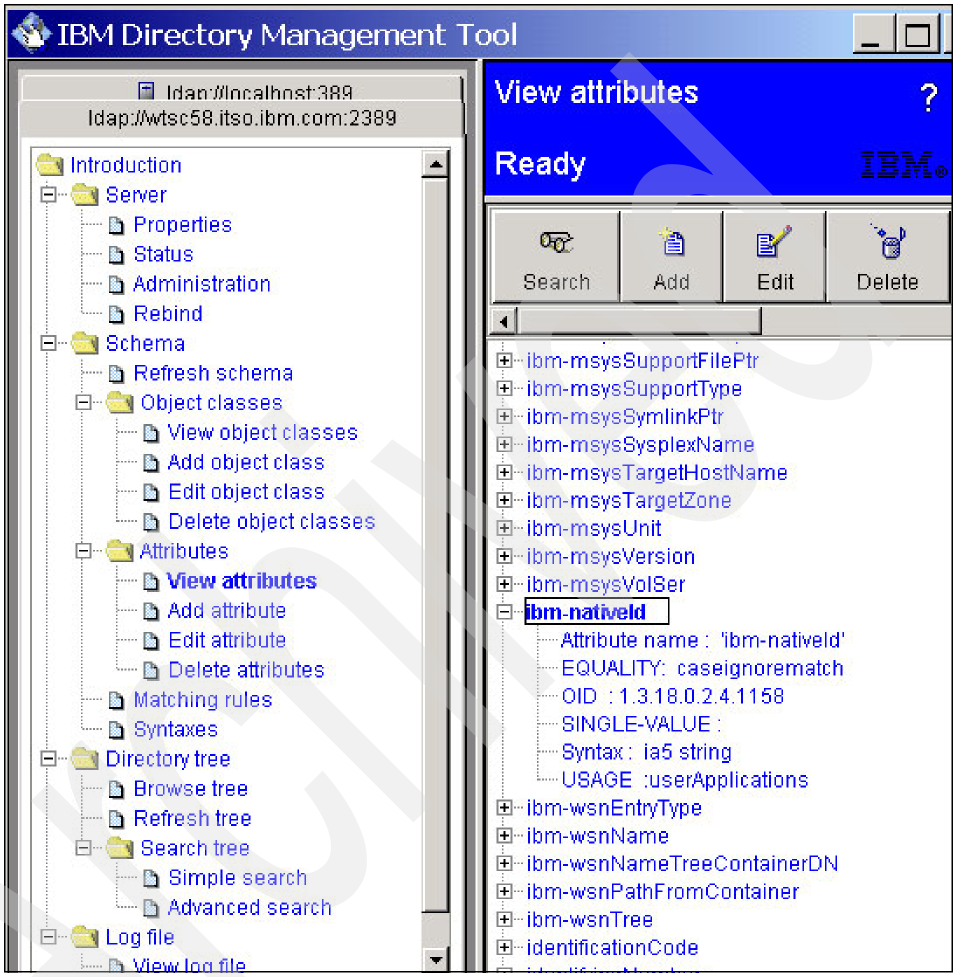


ibm-msysService
ibm-msysServiceLookup
ibm-msysSharedData
ibm-msysSMPEInfo
ibm-msysSoftwareElement
ibm-msysSoftwareFeature
ibm-msysSysplex
ibm-msysVSAMCluster
ibm-nativeAuthentication
Required attributes
ibm-nativeId
Object class name : 'ibm-nativeAuthentica
Object class type : AUXILIARY
OID : 1.3.18.0.2.6.264
Superior object class : top
ibm-SecurityIdentities
ibmsubschema
ibm-wsnEntry
ibm-wsnNameTreeContainer
ibm-wsnPrimaryContextLocation
iGNObject
iGNPerson
INamingService

Figure 7-15 Viewing the installed object classes

5. You can view the details of the attributes by navigating down to Schema->Attributes->View attributes in the navigation pane.

6. In the main window search for `ibm-nativeId` and expand it by clicking on the plus icon. From Figure 7-16 we can see this attribute is of String type and its value is not case-sensitive.



```
ibm-msysSupportFilePtr
ibm-msysSupportType
ibm-msysSymlinkPtr
ibm-msysSysplexName
ibm-msysTargetHostName
ibm-msysTargetZone
ibm-msysUnit
ibm-msysVersion
ibm-msysVolSer
ibm-nativeId
    Attribute name : 'ibm-nativeId'
    EQUALITY: caseignorematch
    OID : 1.3.18.0.2.4.1158
    SINGLE-VALUE :
    Syntax : ia5 string
    USAGE :userApplications
ibm-wsnEntryType
ibm-wsnName
ibm-wsnNameTreeContainerDN
ibm-wsnPathFromContainer
ibm-wsnTree
identificationCode
```

Figure 7-16 Viewing the ibm-nativeId attribute

7. If you click on **Browse** tree in the navigation pane you will see the LDAP tree. Expand the suffix that is in use for WebSphere Portal on z/OS and OS/390 and you should see something similar to what is shown in Figure 7-17 on page 307. Note, our suffix was `dc=ibm,dc=com`.

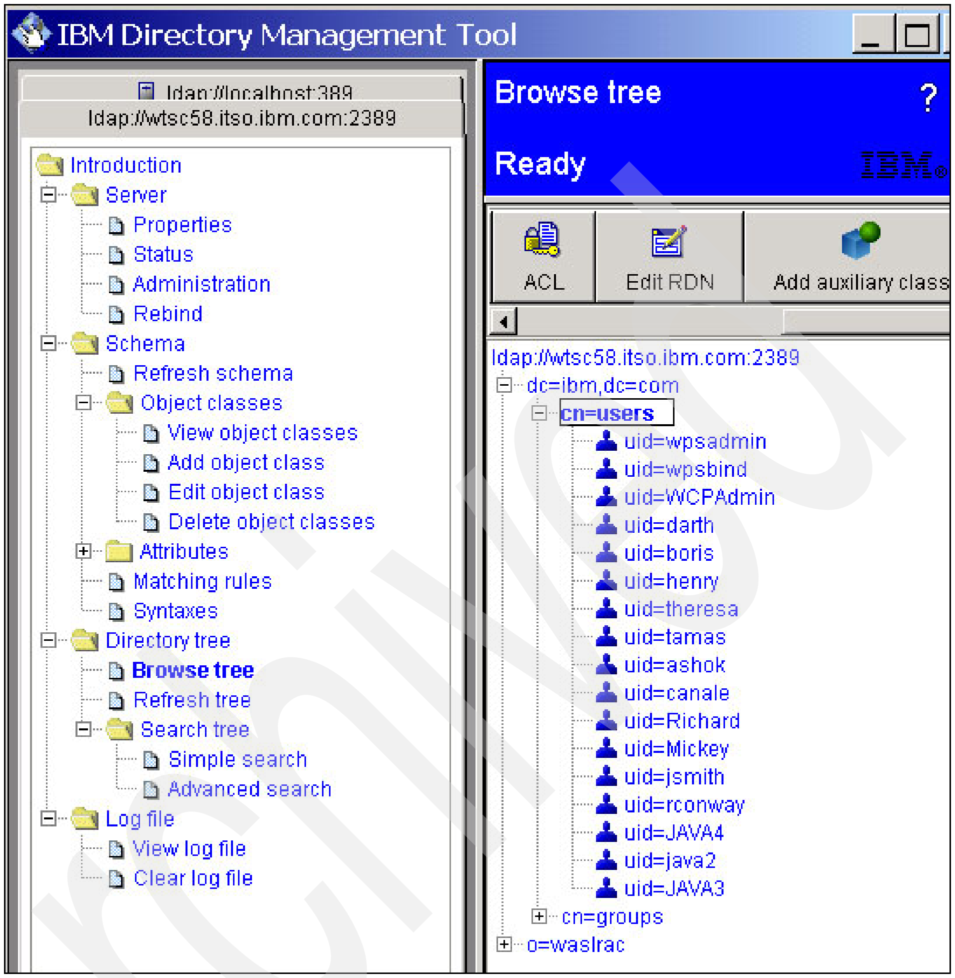
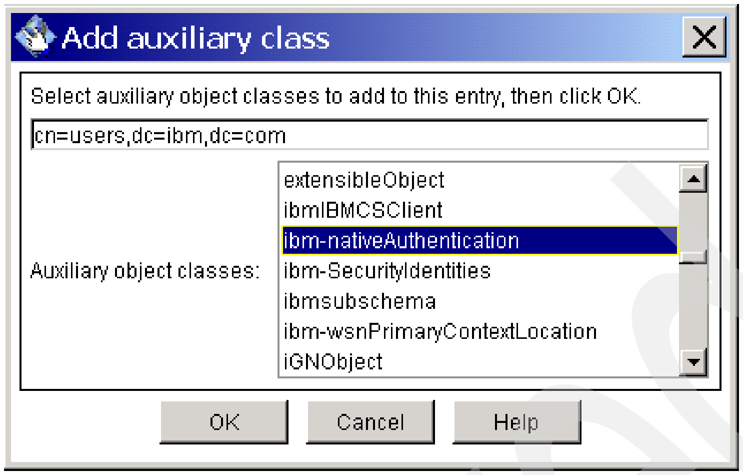


Figure 7-17 Browsing the LDAP tree used by WebSphere Portal on z/OS

8. Before you can add the attribute to the user, that attribute should be available at the higher Root DN (RDN™). In this case the ibm-nativeId attribute should first be added to `cn=users`.
9. Highlight `cn=users` and click **`Add auxiliary class`**. In the pop-up window, select **`ibm-nativeAuthentication`** and click **`OK`** as shown in Figure 7-18 on page 308.



Add auxiliary class

Select auxiliary object classes to add to this entry, then click OK.

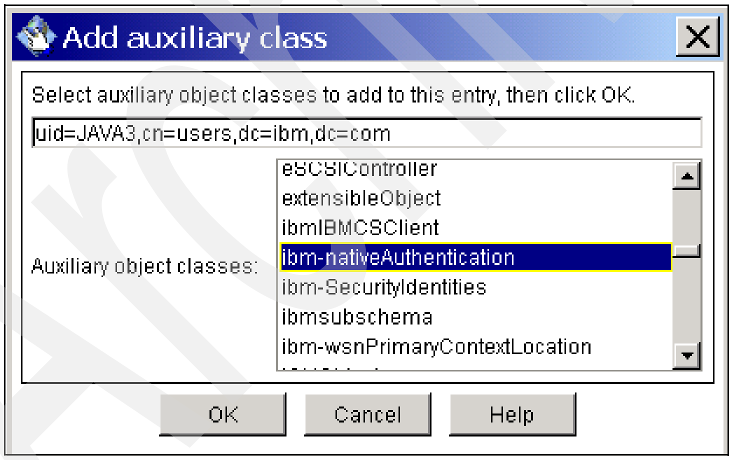
cn=users,dc=ibm,dc=com

Auxiliary object classes:

- extensibleObject
- ibmIBMCSClient
- ibm-nativeAuthentication
- ibm-SecurityIdentities
- ibmsubschema
- ibm-wsnPrimaryContextLocation
- iGNObject

Figure 7-18 Adding auxiliary class at the cn=users level

10. Now add that class to a LDAP user. In a similar manner highlight the user and click **`Add auxiliary class`**. In the pop-up window, select **`ibm-nativeAuthentication`** and click **`OK`**. We chose `uid=JAVA3` as shown in Figure 7-19.



Add auxiliary class

Select auxiliary object classes to add to this entry, then click OK.

uid=JAVA3,cn=users,dc=ibm,dc=com

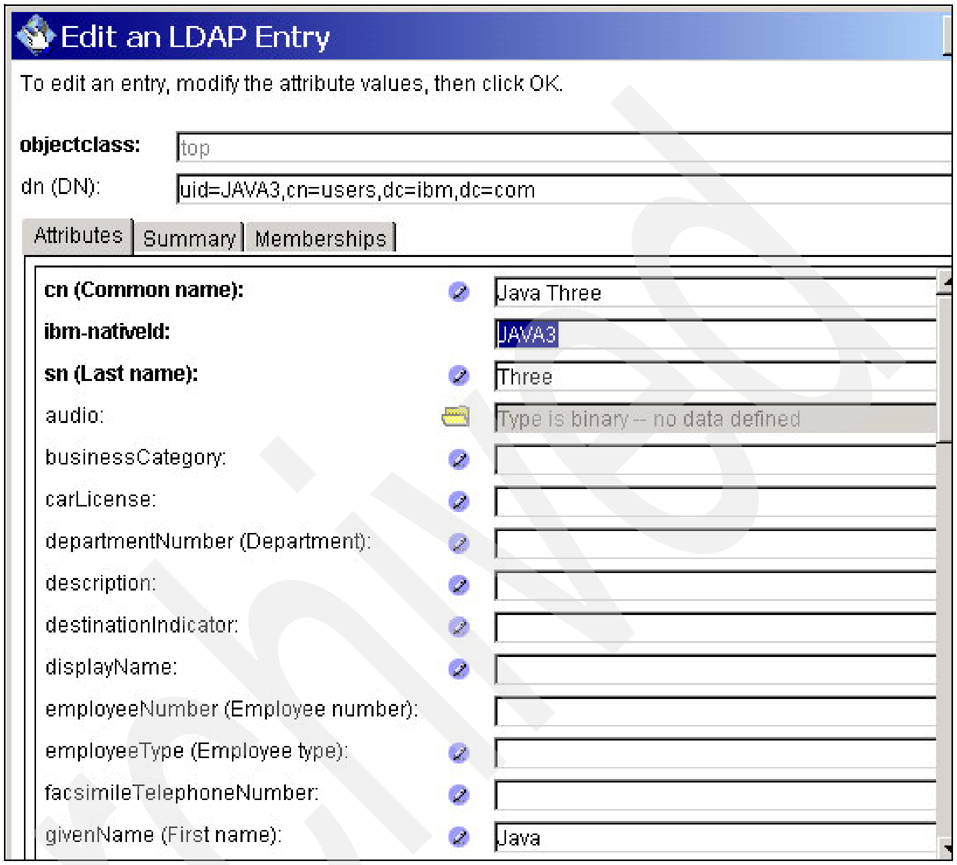
Auxiliary object classes:

- eSCSIController
- extensibleObject
- ibmIBMCSClient
- ibm-nativeAuthentication
- ibm-SecurityIdentities
- ibmsubschema
- ibm-wsnPrimaryContextLocation

Figure 7-19 Adding auxiliary class at the UID level

11. If you examine the attribute details of the user id that was modified you will see a new required attribute called ibm-nativeId. You can view the attributes by highlighting **`uid=JAVA3`** in the main window of DMT and clicking **`Edit`**. Notice the label is displayed in bold, indicating that this is a required field.

12. The value of this attribute should be the RACF user id. Enter the RACF UID and click **OK**. In our case the RACF UID was also JAVA3. See Figure 7-20.


Edit an LDAP Entry

To edit an entry, modify the attribute values, then click OK.

objectclass:	top
dn (DN):	uid=JAVA3,cn=users,dc=ibm,dc=com

cn (Common name):	Java Three
ibm-nativeId:	JAVA3
sn (Last name):	Three
audio:	Type is binary -- no data defined
businessCategory:	
carLicense:	
departmentNumber (Department):	
description:	
destinationIndicator:	
displayName:	
employeeNumber (Employee number):	
employeeType (Employee type):	
facsimileTelephoneNumber:	
givenName (First name):	Java

Figure 7-20 Adding the RACF UID value in the ibm-nativeID field

13. The next step is to verify that all the modifications done to enable LDAP native authentication actually work.

7.5.4 Solution verification

Bring up the WebSphere Portal login page via the URL

```
http://<HOST_NAME>/wps/myportal
```

Enter the User ID and Password and make sure you can authenticate successfully. Interestingly, when you log into WebSphere Portal successfully there is no way to tell whether the authentication was done by the LDAP server or by RACF. Therefore, to verify that the native authentication is truly setup correctly, you have to run a negative test.

Enter the User ID of a user that exists in RACF. Enter a wrong password and attempt to login. You should not be successful. If you examine the system log using z/OS System Display and Search Facility (SDSF) you should see a message from RACF rejecting the authentication. That tells us that the userid and password was passed on to RACF because of the special LDAP attribute, namely ibm-nativeId, and indeed RACF performed the authentication check but returned a failure.

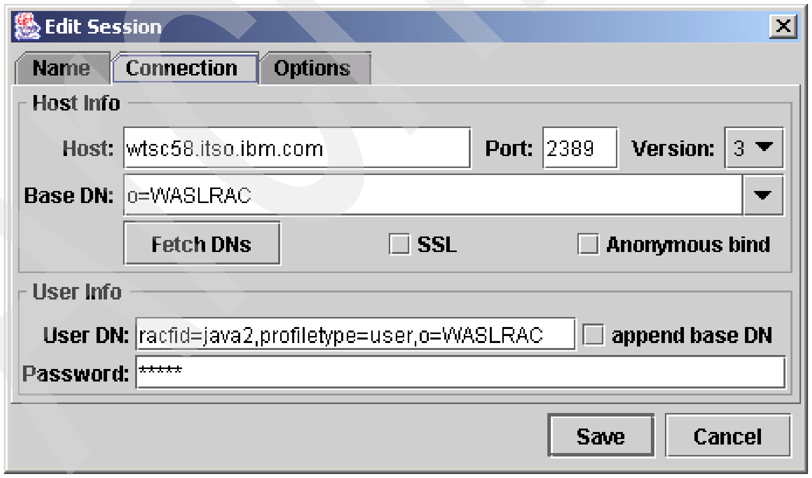
That successfully demonstrates the setup and verification of Native Authentication.

7.5.5 Tips for exporting RACF users to LDAP

Here we provide sample solutions to add RACF users to the LDAP database.

Accessing RACF via an LDAP browser

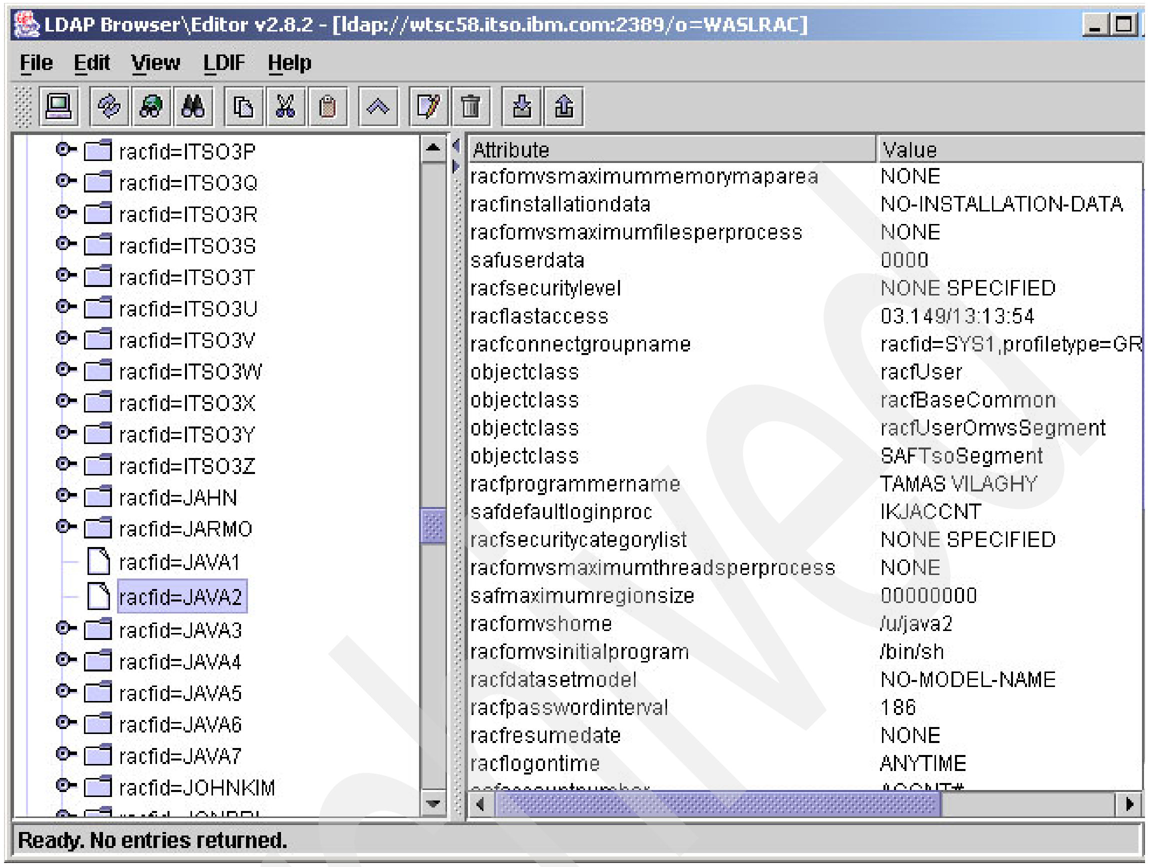
Users in RACF can be exported to an LDAP LDIF file by using an LDAP client. For example we used a freely available LDAP Browser with an LDIF export capability. To use the LDAP client and access RACF you need to sign on with the following DN format: `racfid=<userid>,profiletype=user,o=WASLRAC` as shown in Figure 7-21.



Host: wtsc58.itso.ibm.com
Port: 2389
Version: 3
Base DN: o=WASLRAC
☐ SSL
☐ Anonymous bind
User DN: racfid=java2,profiletype=user,o=WASLRAC
☐ append base DN
Password: *****

Figure 7-21 Accessing RACF via an LDAP client

Figure 7-22 on page 311 shows the LDAP client display of users in RACF.


LDAP Browser\Editor v2.8.2 - [ldap://wtsc58.itso.ibm.com:2389/o=WASLRAC]
File
Edit
View
LDIF
Help

racfid=ITSO3P
racfid=ITSO3Q
racfid=ITSO3R
racfid=ITSO3S
racfid=ITSO3T
racfid=ITSO3U
racfid=ITSO3V
racfid=ITSO3W
racfid=ITSO3X
racfid=ITSO3Y
racfid=ITSO3Z
racfid=JAHN
racfid=JARMO
racfid=JAVA1
racfid=JAVA2
racfid=JAVA3
racfid=JAVA4
racfid=JAVA5
racfid=JAVA6
racfid=JAVA7
racfid=JOHNKIM

Attribute	Value
racfomvsmaximummemorymaparea	NONE
racfinstallationdata	NO-INSTALLATION-DATA
racfomvsmaximumfilesperprocess	NONE
safuserdata	0000
racfsecuritylevel	NONE SPECIFIED
racflastaccess	03.149/13:13:54
racfconnectgroupname	racfid=SYS1,profiletype=GR
objectclass	racfUser
objectclass	racfBaseCommon
objectclass	racfUserOmvsSegment
objectclass	SAFTsoSegment
racfprogrammername	TAMAS VILAGHY
safdefaultloginproc	IKJACCNT
racfsecuritycategorylist	NONE SPECIFIED
racfomvsmaximumthreadsperprocess	NONE
safmaximumregionsize	00000000
racfomvshome	/u/java2
racfomvsinitialprogram	/bin/sh
racfdatasetmodel	NO-MODEL-NAME
racfpasswordinterval	186
racfresumedate	NONE
racflogontime	ANYTIME

Ready. No entries returned.

Figure 7-22   RACF users displayed in an LDAP client

After using the LDAP client to export all the users to an LDIF file the format, the partial contents of the file is shown in Example 7-4.

Example 7-4   Example of LDIF export from RACF

```
dn: racfid=ALEX,profiletype=USER, o=WASLRAC

dn: racfid=ALEX1,profiletype=USER, o=WASLRAC

dn: racfid=ALEX2,profiletype=USER, o=WASLRAC

dn: racfid=ALVAREZ,profiletype=USER, o=WASLRAC

dn: racfid=AMCHENG,profiletype=USER, o=WASLRAC

dn: racfid=ANDY,profiletype=USER, o=WASLRAC
```

```
dn: racfid=ANTOGNI,profiletype=USER, o=WASLRAC

dn: racfid=ANTONIO,profiletype=USER, o=WASLRAC

dn: racfid=ARDINI,profiletype=USER, o=WASLRAC

dn: racfid=ARMIGES,profiletype=USER, o=WASLRAC
```

This format, with a single line for each user is quite easy to modify using a variety of tools. The only part we are interested in is the userid's identified by field `racfid`. We need to identify which users we do want to add to the Portal Server LDAP for native authentication and discard all other users. A suggested way to parse and modify the file are:

- Use macros in a word processor
- User macros in a spreadsheet
- Use a script, for example REXX or UNIX shell
- If the number of users is small, use an LDAP editor

The LDIF format exported needs to be parsed as suggested and converted into the LDIF format used by Portal Server, with its required attributes. As an example of this we added two users. CHEN and JAVA7 to an LDIF file, we called `NAUpdate.ldif` as shown in Example 7-5. Note that the required format does not use or need an attribute for the `userpassword`.

Example 7-5 NAUpdate.ldif, LDIF file for portal using native authentication

```
dn: uid=CHEN,cn=users, dc=ibm,dc=com
givenname: Chen
ibm-nativeid: CHEN
sn: Chen
mail: chen@uk.ibm.com
objectclass: top
objectclass: inetOrgPerson
objectclass: Person
objectclass: ORGANIZATIONALPERSON
objectclass: ibm-nativeAuthentication
uid: CHEN
cn: Chen
preferredlanguage: en

dn: uid=JAVA7,cn=users, dc=ibm,dc=com
givenname: Java7
ibm-nativeid: JAVA7
sn: Java7
```

```
mail: java7@us.ib.com
objectclass: top
objectclass: inetOrgPerson
objectclass: Person
objectclass: ORGANIZATIONALPERSON
objectclass: ibm-nativeAuthentication
uid: JAVA7
cn: Java7
preferredlanguage: en
```

In our example user CHEN was not in RACF, but user JAVA7 was in RACF. We added the users to the Portal server LDIF by executing the following command:

```
ldapmodify -a -p 2389 -f ./NAUpdate.ldif -D “cn=CBAdmin” -w
<password>
```

After this we tested that the changes were successful as follows:

We tried to login to Portal server using userid CHEN. This did not succeed due to an authentication error because userid CHEN was not in RACF.

We logged in successfully to Portal server using userid JAVA7, which was in RACF, and confirms the successful modification of LDAP with users authenticating via native authentication.

7.5.6 Mass-exporting RACF users to LDAP

If you are planning on exporting a large amount of RACF userid(s) into LDAP server as registered Portal users, use the following example.

Note: The mass-exporting RACF users to LDAP is not a requirement for Portal Server configuration and installation. This can be part of initial environment set up for a corporate Portal. Instead of having all employees to self-register or the RACF, LDAP administrators to register all individual employees as portal users.

The mass-exporting RACF users to LDAP requires LDAP Native Authentication.

Exporting from RACF

First, here are the steps for exporting your RACF native userids to a file using the LDAP Browser which has an LDIF export capability.

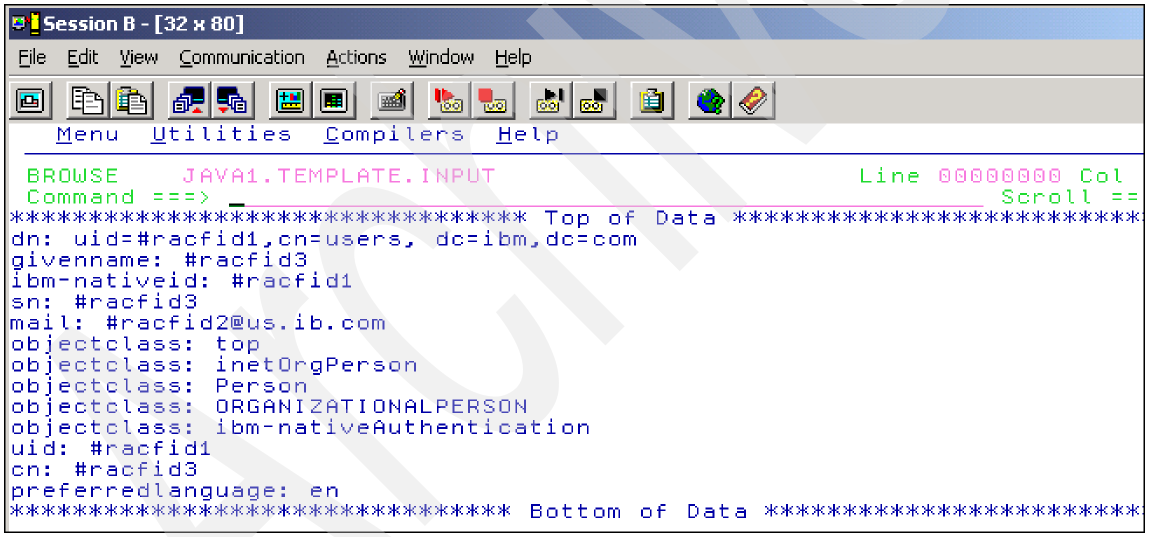
- You need to sign on to the LDAP Browser with a valid RACFID as shown in Figure 7-21 on page 310.

- Move your mouse to the objectclass **"profiletype=user"** and click on it. Move your mouse to the browser's menu **"LDIF"** and select **"Export"**.
- Click the "**All children**" option; expand the **"…"** folder option on the upper-right corner and select your desired folder and enter the file name.
- Click the **"Export"** button and the file will be generated.

Running the REXX program

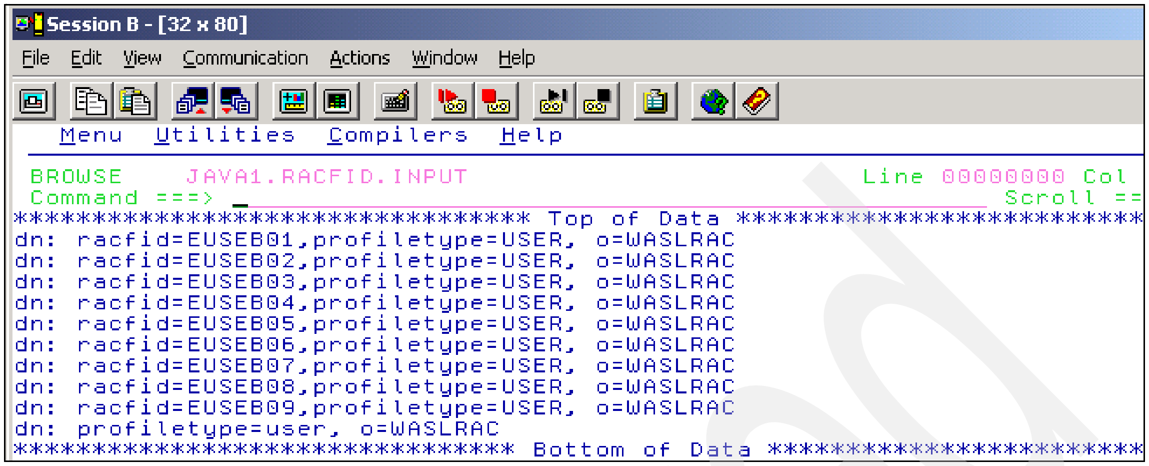
Second, we have prepared and tested a sample REXX program and a template file with predefined tokens similar to Example 7-5 on page 312 for you to use. Please refer to Appendix D, "Additional material" on page 339 for details how to get the files. You will need to unzip the downloaded file. The zip file contains:

- A REXX program convldif.rexx (see Example C-1 on page 331).
- A template file template.ldif (see Figure 7-23).
- A sample exported RACF users list file (see Figure 7-24 on page 315).
- A sample merged output file merged.output. (see Figure 7-25 on page 315).


Session B - [32 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

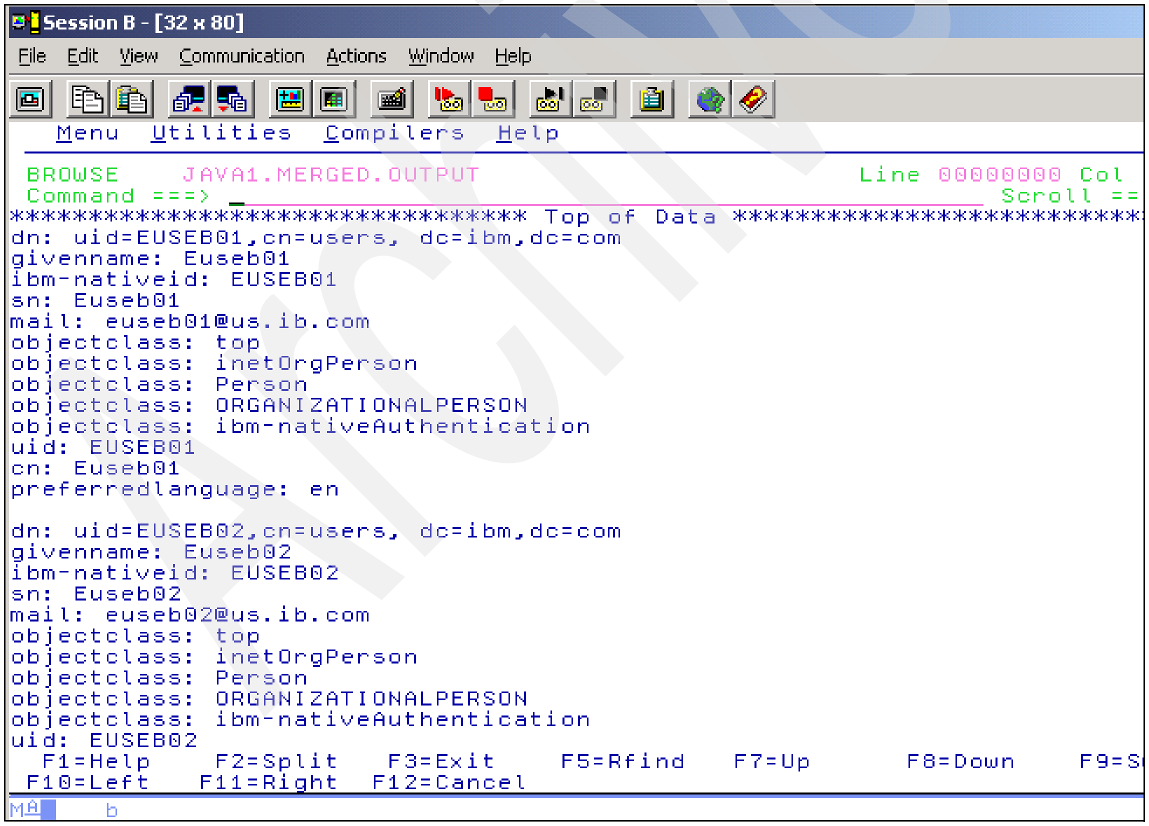
```
 BROWSE    JAVA1.TEMPLATE.INPUT                         Line 00000000 Col
 Command ===> _                                                Scroll ==
********************************* Top of Data **************************
dn: uid=#racfid1,cn=users, dc=ibm,dc=com
givenname: #racfid3
ibm-nativeid: #racfid1
sn: #racfid3
mail: #racfid2@us.ib.com
objectclass: top
objectclass: inetOrgPerson
objectclass: Person
objectclass: ORGANIZATIONALPERSON
objectclass: ibm-nativeAuthentication
uid: #racfid1
cn: #racfid3
preferredlanguage: en
******************************** Bottom of Data ************************
```

Figure 7-23 Sample LDIF template


Session B - [32 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

```
 BROWSE    JAVA1.RACFID.INPUT                              Line 00000000 Col
 Command ===> _                                                   Scroll ==
********************************* Top of Data **************************
dn: racfid=EUSEB01,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB02,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB03,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB04,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB05,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB06,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB07,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB08,profiletype=USER, o=WASLRAC
dn: racfid=EUSEB09,profiletype=USER, o=WASLRAC
dn: profiletype=user, o=WASLRAC
******************************** Bottom of Data ************************
```

Figure 7-24 The exported RACFID sample file


Session B - [32 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

```
 BROWSE    JAVA1.MERGED.OUTPUT                             Line 00000000 Col
 Command ===> _                                                   Scroll ==
********************************* Top of Data **************************
dn: uid=EUSEB01,cn=users, dc=ibm,dc=com
givenname: Euseb01
ibm-nativeid: EUSEB01
sn: Euseb01
mail: euseb01@us.ib.com
objectclass: top
objectclass: inetOrgPerson
objectclass: Person
objectclass: ORGANIZATIONALPERSON
objectclass: ibm-nativeAuthentication
uid: EUSEB01
cn: Euseb01
preferredlanguage: en

dn: uid=EUSEB02,cn=users, dc=ibm,dc=com
givenname: Euseb02
ibm-nativeid: EUSEB02
sn: Euseb02
mail: euseb02@us.ib.com
objectclass: top
objectclass: inetOrgPerson
objectclass: Person
objectclass: ORGANIZATIONALPERSON
objectclass: ibm-nativeAuthentication
uid: EUSEB02
  F1=Help      F2=Split     F3=Exit      F5=Rfind     F7=Up        F8=Down      F9=S
 F10=Left     F11=Right    F12=Cancel
MA       b
```

Figure 7-25 Merged sample LDIF output file

You will then need to upload all these four files to the z/OS mainframe area.

Logon to TSO and use ISPF 3.2 to create three new datasets:

- “<YourHLQ>.RACFID.INPUT” (RACF userid list file)
- “<YourHLQ>.TEMPLATE.INPUT” (Template ldif input file)
- “<YourHLQ>.MERGED.OUTPUT”(Merged ldif output file).

The RACF users and the template input datasets should have PS Organization, RECFM=FB and record length of 80, whereas the merged output dataset should have PS Organization, RECFM=VB and Record Length of 255.

These three files are used by the sample REXX program convldif.rexx. Also copy the REXX program convldif.rexx to your Portal Server Installation library (that is, “<YourHLQ>.JOBS.CNTL”) as the member CONVLDIF. Do the following customization:

- The REXX program CONVLDIF
  Replace all three <YourHLQ> to match with your selected dataset names. See Example C-1 on page 331.
- <YourHLQ>.RACFID.INPUT
  This input file contains the begin pattern of **‘racfid=’** and the end pattern **‘,’** that the REXX program CONVLDIF will extract any names specified in between the two patterns.
- <YourHLQ>.TEMPLATE.INPUT
  The template file contains three recognized token names to be replaced by the extracted RACF userid names.
  - #racfid1 is the default userid name in uppercase
  - #racfid2 is the userid name in lowercase
  - #racfid3 is the userid name with first letter in upper case

Note: Only one token per line will be replaced with extracted RACF name and any line which doesn’t have those three recognized definitions will be generated as is. You will also need to change the “dc=ibm,dc=com”.

- <YourHLQ>.MERGED.OUTPUT
  After running the REXX exec, this file will contain the generated output in ldif format. Make sure the template file format is valid, so the output file will contain the valid ldif format as well.

To invoke the REXX program, you should logon to TSO, then ISPF and select option 6, “command”. At the ISPF Command Shell prompt, enter “`exec`

<YourHLQ>.JOBS.CNTL(CONVLDIF)" to execute the CONVLDIF REXX program. The merged output should be in the <YourHLQ>.MERGED.OUTPUT dataset.

Importing into LDAP

Once you have modified all the required dataset prefixes and ldif suffix changes, you should be able to execute this REXX program and review the merged output file. After validating the results, the next step is to import the ldif file into your Portal LDAP server using the **ldapmodify** command.

```
ldapmodify -a -h 127.0.0.1 -p <your LDAP port #> -D cn=CBAdmin -w <pswd>
-f <your ldif file name>
```

8

Problem determination

In this chapter we provide problem descriptions and determinations materials that we gathered through our troubleshooting efforts.

8.1 Portlet deployment problems

In this section we have the PDDs for deployment problems.

8.1.1 PDD001D - MSG - WP00027E Creation of conversation failed

The message, `PDD001D - MSG - WP00027E Creation of conversation failed`, is described here.

Phase - Portlet deployment

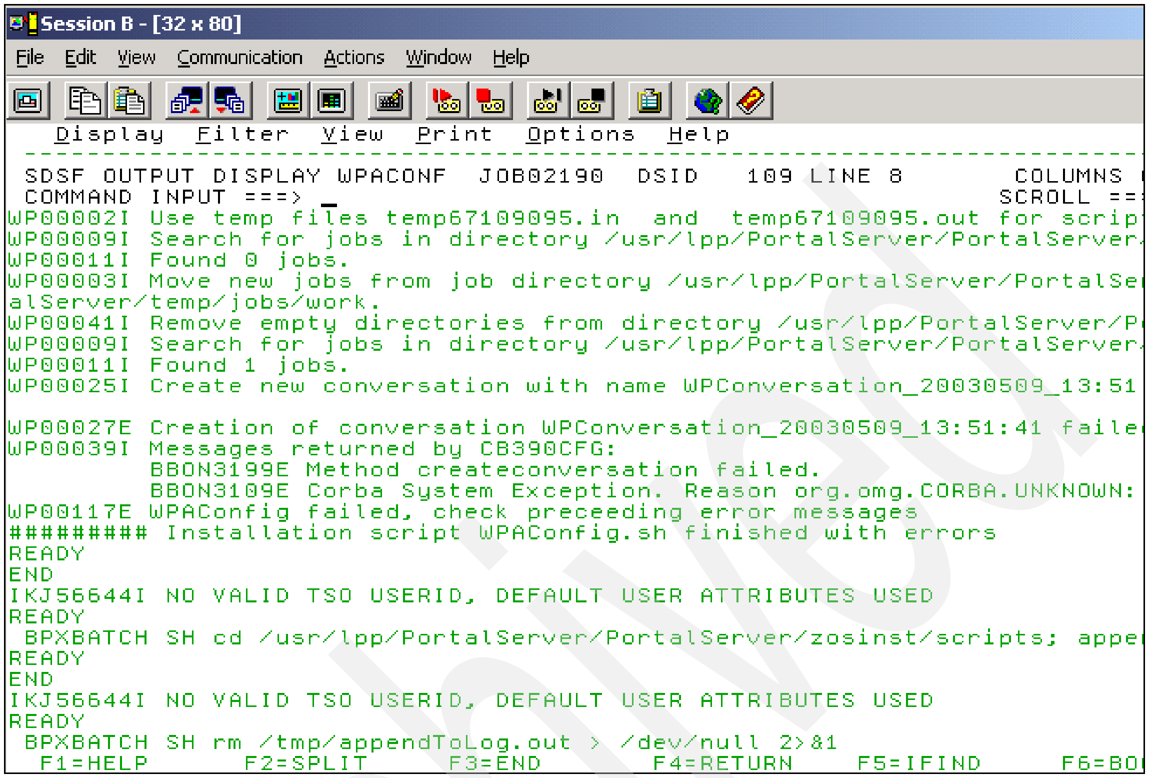
This message may be found in the held output for job WPACONF when deploying portlets added from the Portal Server browser administrative place.

Extended description

Additional messages:

- BB0N3199E Method create failed
- BB0N3109E Corba system exception

An example of the job log output is shown in Figure 8-1 on page 321.



```
   Display  Filter  View  Print  Options  Help
 -----------------------------------------------------------------------
 SDSF OUTPUT DISPLAY WPACONF  JOB02190  DSID   109 LINE 8       COLUMNS
 COMMAND INPUT ===> _                                          SCROLL ==
WP00002I Use temp files temp67109095.in  and  temp67109095.out for scrip
WP00009I Search for jobs in directory /usr/lpp/PortalServer/PortalServer
WP00011I Found 0 jobs.
WP00003I Move new jobs from job directory /usr/lpp/PortalServer/PortalSe
alServer/temp/jobs/work.
WP00041I Remove empty directories from directory /usr/lpp/PortalServer/P
WP00009I Search for jobs in directory /usr/lpp/PortalServer/PortalServer
WP00011I Found 1 jobs.
WP00025I Create new conversation with name WPConversation_20030509_13:51

WP00027E Creation of conversation WPConversation_20030509_13:51:41 faile
WP00039I Messages returned by CB390CFG:
         BBON3199E Method createconversation failed.
         BBON3109E Corba System Exception. Reason org.omg.CORBA.UNKNOWN:
WP00117E WPAConfig failed, check preceeding error messages
######### Installation script WPAConfig.sh finished with errors
READY
END
IKJ56644I NO VALID TSO USERID, DEFAULT USER ATTRIBUTES USED
READY
 BPXBATCH SH cd /usr/lpp/PortalServer/PortalServer/zosinst/scripts; appe
READY
END
IKJ56644I NO VALID TSO USERID, DEFAULT USER ATTRIBUTES USED
READY
 BPXBATCH SH rm /tmp/appendToLog.out > /dev/null 2>&1
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=IFIND     F6=BO
```

Figure 8-1 WPACONF failure Create Conversation failed

Diagnosis and solution

Delete some old and unnecessary conversations using SMEUI.

8.1.2 PDD002D - MSG - IRX0043I Error running WPAConfig, line 524

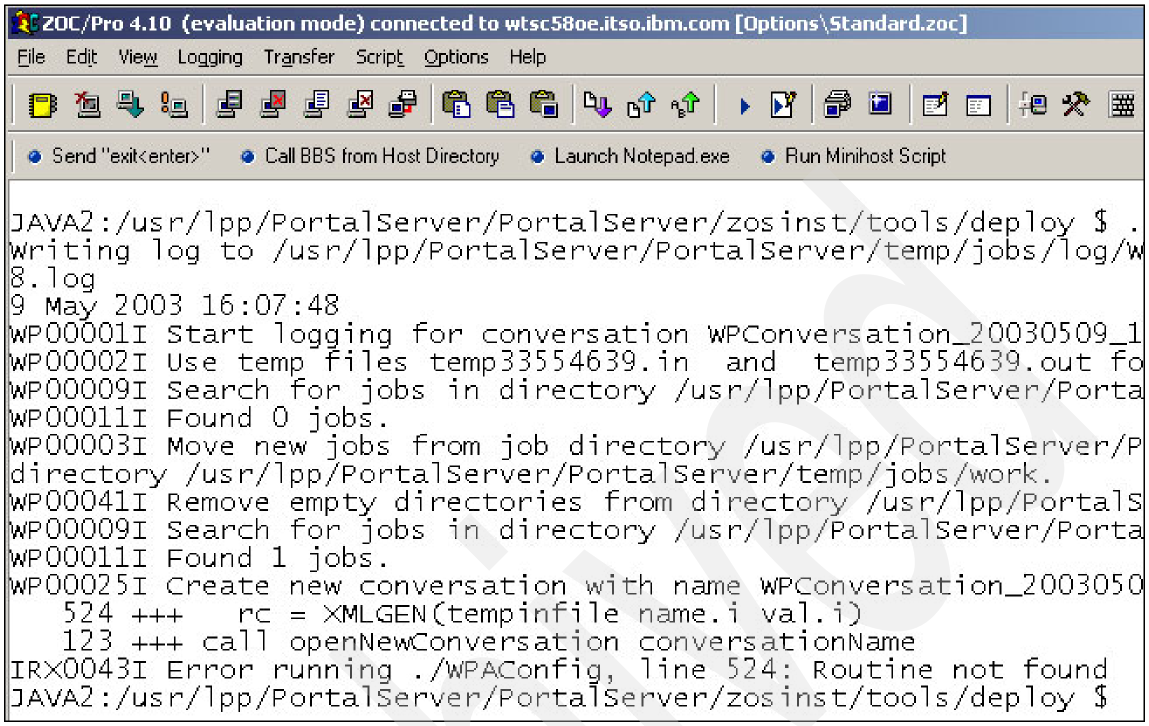
The message, `PDD002D - MSG - IRX0043I Error running WPAConfig, line 524: Routine not found`, is described here.

Phase - Portlet deployment

This message will be seen on the UNIX telnet console after running shell script WPAConfig when deploying portlets added from the Portal Server browser administrative place.

Extended description

An example of the telnet console is shown in Figure 8-2 on page 322.


ZOC/Pro 4.10 (evaluation mode) connected to wtsc58oe.itso.ibm.com [Options\Standard.zoc]
File Edit View Logging Transfer Script Options Help
Send "exit<enter>"
Call BBS from Host Directory
Launch Notepad.exe
Run Minihost Script

```
JAVA2:/usr/lpp/PortalServer/PortalServer/zosinst/tools/deploy $ .
Writing log to /usr/lpp/PortalServer/PortalServer/temp/jobs/log/W
8.log
9 May 2003 16:07:48
WP00001I Start logging for conversation WPConversation_20030509_1
WP00002I Use temp files temp33554639.in  and  temp33554639.out fo
WP00009I Search for jobs in directory /usr/lpp/PortalServer/Porta
WP00011I Found 0 jobs.
WP00003I Move new jobs from job directory /usr/lpp/PortalServer/P
directory /usr/lpp/PortalServer/PortalServer/temp/jobs/work.
WP00041I Remove empty directories from directory /usr/lpp/PortalS
WP00009I Search for jobs in directory /usr/lpp/PortalServer/Porta
WP00011I Found 1 jobs.
WP00025I Create new conversation with name WPConversation_2003050
   524 +++    rc = XMLGEN(tempinfile name.i val.i)
   123 +++ call openNewConversation conversationName
IRX0043I Error running ./WPAConfig, line 524: Routine not found
JAVA2:/usr/lpp/PortalServer/PortalServer/zosinst/tools/deploy $
```

Figure 8-2   WPACONF failure Create Conversation failed

Diagnosis and solution

This error of not able to add/deploy new portlets was caused by insufficient DB2 table space. We first got around the problem by temporarily deleted some old and unneeded conversations under SMEUI. The real solution is to add additional extents, increase the size of the DB2 table space.

8.2 SMEUI problems

In this section we have the PDDs for administration problems.

8.2.1 PDD003I - MSG - BBON0579E Cannot activate conversation

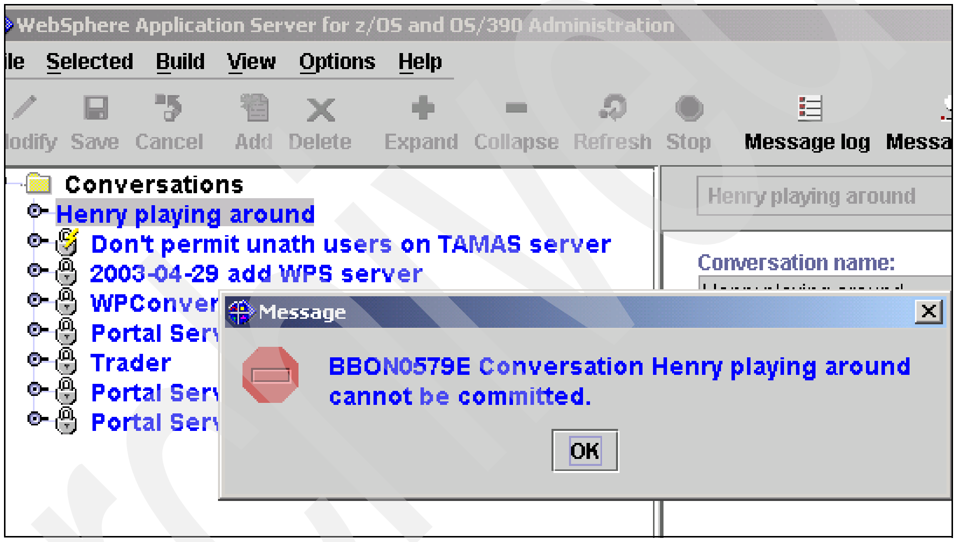
The message, `PDD003I - MSG - BBON0579E Cannot activate conversation`, is described here.

Phase - Changing configuration

This message will show as a pop up error window from SMEUI when trying to Activate a validated conversation.

Extended description

The most likely cause of this error is that another conversation has been created, committed and activated since this one was created. In the case of WebSphere Portal Server on z/OS this may happen if a portlet is deployed using job WPACONF to execute script WPConfig since the time the conversation on SMEUI was created. Your only course of action is to delete the conversation created on SMEUI and re-create it.


WebSphere Application Server for z/OS and OS/390 Administration
ile Selected Build View Options Help
lodify Save Cancel Add Delete Expand Collapse Refresh Stop Message log Messa
Conversations
Henry playing around
Don't permit unath users on TAMAS server
2003-04-29 add WPS server
WPConver
Portal Serv
Trader
Portal Serv
Portal Serv
Henry playing around
Conversation name:
Message
OK

BBON0579E Conversation Henry playing around cannot be committed.

Figure 8-3 SMEUI error BBON0579E

Diagnosis and solution

Remember to coordinate administration/deployment efforts. One conversation at a time.

A

Portal Server environment variables

The Portal server environment variables should be set up and agreed upon prior to the actual installation process. Table A-1 shows the variables that were being set for the redbook project's environment, in **bold**.

Table A-1 WebSphere Portal Server environment variables

Variable Name	Description	Required	Sample Value
WPS_CONVNAME	Name of the conversation used for installation of WebSphere Portal	No	WebSphere Portal 1
WPS_SERVERNAME	Server name of WebSphere Portal, at most 7 characters long	YES	**WSPORT**
WPS_CRLEGIONID	Name of the server region userID of WebSphere Portal. This name must be setup in RACF	YES	**WSPORTCU**
WPS_SRVREGIONID	Name of the server regionuser ID of WebSphere Portal. This name must be setup in RACF	YES	**WSPORTSU**
WPS_GROUP	Name of the Group of WebSphere Portal	YES	**WSPORTS**

Variable Name	Description	Required	Sample Value
WPS_LOGSTREAM	Log stream name of WebSphere Portal	No	**WAS.ERROR.LOG**
WPS_PROCNAME	Procedure name of the control region of WebSphere Portal. This name be setup in PROCLIB	YES	**WSPORT**
WPS_PATH	Installation path of WebSphere Portal	YES	**/usr/lpp/PortalServer /PortalServer**
WPS_LOG_PATH	Log file path of WebSphere Portal	No	**/usr/lpp/PortalServer**
WPS_LOG_File	Log file name of WebSphere Portal	No	**wps.log**
WPS_PORT	HTTP port of WebSphere Portal	Yes	**8082**
WPS_NUM_SERVER INSTANCES	Number of server instances per system created for WebSphere Portal	No	**2**
WPS_INSTALL_ SYSTEMS	List of systems on which server instances for WebSphere Portal should be created. If default, server instances are created on all systems	No.	
WPS_RESORUCE_ID	UserID for the DB2 datasource used by WebSphere Portal. This user ID must have the DB2 authority to create database tables.	YES	**JAVA3** (this is the TSO user session ID we used in the redbook project environment)
WPS_RESOURCE_W	Password for the DB2 datasource used by WebSphere Portal.	YES	**JAVA3** (this is the TSO user session ID we used in the redbook project environment)
WPS_RESOURCE_ LOCATION	Location name of the database to be used when establishing connections with the DB2 datasource used by WebSphere Portal.	No	COM4D741
WPS_HOST	Fully-Qualified name of the WebSphere Portal host. (host name and domain name in lower case)	YES	**wtsc58.itso.ibm.com**
WPS_SHORT_HOST	Short name of WebSphere Portal host	YES	**wtsc58**

Variable Name	Description	Required	Sample Value
WPS_PBULIC_HOME	URI of the public area of WebSphere Portal	YES	portal
WPS_PROTECTED_ HOME	URI of the protected area of WebSphere Portal	YES	myportal
WPS_PROXY-HOST	Fully-qualified name of the proxy server used by WebSphere Portal (Host name and Domain name)	YES	proxy.there.everywher e
WPS_PROXY_PORT	Port number of the proxy server used by WebSPhere Portal	YES	80
WPS_LDAP_SERVER	Fully-qualified name of the LDAP server used by WebSphere Portal (Host name and Domain name)	YES	**wtsc58.itso.Ibm.com**
WPS_LDAP_PORT	Port number of the LDAP server used by WebSphere Portal	No	**2389**
WPS_LDAP_ID	LDAP user ID use by WebSphere Portal	YES	uid-wpsbind,cn=users, **dc=ibm,dc=com**
WPS_LDAP-PW	LDAP password used by WebSphere Portal	YES	**wpsbind**
WTP-PATH	Installation path of WebSphere Transcoding Publisher	YES	**/usr/lpp/PortalServer /IBMTrans**
WCP_PATH	Installation path of WebSphere Portal Content Publishing	YES	**/usr/lpp/PortalServer /wcp**
WS-PATH	Installation path of WebSphere Application Server on z/OS or OS/390	YES	**/usr/lpp/WebSphere**
WS-CONFIG_PATH	Configuration path of WebSphere Application Server z/OS or OS/390	YES	**/WebSphere390/CB3 90**
DB2-PATH	Installation path of DB2	YES	**/usr/lpp/db2/db2710**
DB2_PROPS	File name of DB2 properties file for JDBC	YES	**/etc/db2sqljjdbc.prop erties**
JAVA_PATH	Installation path of the JDK	YES	**/usr/lpp/java/ibm/J1. 3**
JVM_LOG	File name of JVM log file	YES	**/tmp/jvm.log**

Variable Name	Description	Required	Sample Value
WPS_RESOURCE	Name of the DB2 datasource used by WebSphere Portal. This is extracted automatically from /WPX_PATH/libapp/config/services/ DataStoreService.properties **Do NOT edit this name**	YES	**wps40DS**

B

WPCP environment variables

The Portal Content Publishing environment variables should be set up and agreed upon prior to the actual installation process. Table B-1 shows the variables that were being set for the redbook project's environment, in **bold**.

Table B-1 WebSphere Portal Content Publishing environment variables

Variable Name	Description	Required	Sample Value
WCP_CONVNAME	Name of the conversation used for configuration of WebSphere Portal Content Publishing	No	Portal content publishing
WCP_PATH	Installation path of WebSphere Portal Content Publishing	YES	**/usr/lpp/PortalServer /wcp**
WCP_PUB_DIR	Installation path of the *publish* directory of WebSphere Portal Content Publishing. Must be the same directory as that entered in the script. *installwcp.sh*	YES	**/PortalServer/publish**

Variable Name	Description	Required	Sample Value
WCP_RESORUCE_ID	User ID for the DB2 datasource used by WebSphere Portal Content Publishing	YES	**JAVA3** (this is the TSO user session ID we used in the redbook project environment)
WCP_RESOURCE_PW	Password for the DB2 datasource used by WebSphere Portal Content Publishing.	YES	**JAVA3** (this is the TSO user session ID we used in the redbook project environment)
WCP_RESOURCE_ LOCATION	Location name of the database to be used when establishing connections with the DB2 datasource used by WebSphere Portal Content Publishing.	No	**DSN7LOC1**
WCP_RESOURCE	Name of the DB2 datasource used by WebSphere Portal Content Publishing. **Do NOT edit this name**	YES	**persDS**
WCP_RESOURCE_DEM O	Name of the DB2 datasource used by WebSphere Portal Content Publishing. **Do NOT edit this name**	YES	**wcpdemo**

C RACF userid conversion script

The following REXX exec was used in 7.5.6, "Mass-exporting RACF users to LDAP" on page 313 to convert exported RACF userids to LDAP entries.

Example: C-1 CONVLDIF REXX program listing

```
/**************************** REXX ************************************/
/* CONVLDIF: Merge an exported RACF user list file and a LDIF       */
/*           template file to an ldif file for Portal Server        */
/*                                                                  */
/********************************************************************/
/* Usage:                                                           */
/*                                                                  */
/* 1. Use an LDIF Browser to export RACF userds to a file. The file  */
/*    will contains dn: racfid=USERXXX,profiletype=USER, o=YYYYYY.   */
/* 2. Create a templete file with correct ldif format.              */
/* 3. Replace the name specified in uid of dn:, givenname,          */
/*    ibm-natived, sn, name portion of the mail, uid and cn fileds   */
/*    with #racfid1, #racfid2 or #racfid3.                          */
/*    #racfid1: Uppercase, #racfid2: lowercase, and #racfid3: first  */
/*    letter uppercase.                                             */
/* 4. Create an empty output file.                                  */
/* 5. Match all the three file names with this REXX script.         */
/*  . Replace <YourHLQ> to your selected datasets pefix (ie. JAVA1)  */
/* 6. Run this REXX script and the outpit file will have the merged  */
/*    ldif format.                                                  */
/*                                                                  */
/********************************************************************/
/*  Date    who      COMMENT                                        */
/* ------- -----  ------------------------------------------------- */
/* 5/29/03  CCC   Created                                           */
/*                                                                  */
/********************************************************************/

/*  step1:  Get input file names                                    */

 "free fi(INDD1)"
 "free fi(INDD2)"
 "free fi(OUTDD1)"

 infile1  = "<YourHLQ>.TEMPLATE.INPUT"
 infile2  = "<YourHLQ>.RACFID.INPUT"
 outfile1 = "<YourHLQ>.MERGED.OUTPUT"

 "alloc fi(INDD1) da('"infile1"') shr reuse"
 "alloc fi(INDD2) da('"infile2"') shr reuse"
 "alloc fi(OUTDD1) da('"outfile1"') shr reuse"

 EOFFLAG1 = 2              /* return code to indicate end-of-file    */
 RETURN_CODE1 = 0          /* initialize return code                 */

"EXECIO 0 DISKR INDD1 (OPEN"      /* open template file           */
"EXECIO 0 DISKR INDD2 (OPEN"      /* open table                   */
"EXECIO 0 DISKW OUTDD1 (OPEN"     /* open outout file             */

/*---------------------------------------------------------------*/
/*  step2:  read in the template file                            */
/*---------------------------------------------------------------*/

 bPattern = 'racfid='                   /* Begin pattern         */
 ePattern = ','                         /* End pattern           */
 token = '#racfid'                      /* Token to be replaced  */

 nTotal = 0             /* Total template lines  */
 lSeg.  = ''            /* Left  Segment of the template line */
 rSeg.  = ''            /* Right segment of the template line */
 lineMerge. = 0         /* Default to no merger for this line */
 Name_case. = 1         /* set default to upper case */
 junk   = ''; s1 =''; s2= '';  s3= ''
 Case   = 0                             /* Case = 1, 2, or 3 */

 strUpper = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  /* Translate table in  */
 strLower = 'abcdefghijklmnopqrstuvwxyz'  /* Translate table out */
```

```
    N = 0                  /* template line counter */
    P = 0                  /* #racfid  location     */
    ALINE = ''             /* init */

say ''
say " *** Start to read template list file ...  "
say ''

/*-----------------------------------------------------------------*/
nTotal = 0            /* Total template lines  */
lSeg.  = ''           /* Left  Segment of the template line */
rSeg.  = ''           /* Right segment of the template line */
lineMerge. = 0        /* Default to no merger for this line */
Name_case. = 1        /* set default to upper case */

junk   = ''; s1 =''; s2= '';  s3= ''
Case   = 0                           /* Case = 1, 2, or 3 */

token1= 'racfid='                       /* locate racfid=      */
token = '#racfid'                       /* Replace with racfid */
strUpper = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  /* Translate table in  */
strLower = 'abcdefghijklmnopqrstuvwxyz'  /* Translate table out */

   N = 0                  /* template line counter */
   P = 0                  /* #racfid  location     */
   ALINE = ''             /* init */

say ''
say " Start to read template list file ...  "
say ''

DO WHILE (RETURN_CODE1 \= EOFFLAG1) /* loop while not end-of-file    */
  'EXECIO 1 DISKR INDD1'          /* read 1 line from template     */
  RETURN_CODE1 = RC               /* save execio rc                */

   IF RETURN_CODE1 = 0 THEN        /* get a line ok?       */
     DO                            /* yes                  */
       N = N + 1                   /* Line counter         */
       PARSE PULL ALINE
       P = POS(TOKEN,ALINE,1)      /* Get TOKEN location */

       IF (P \= 0) THEN            /* IF a token is found */
          DO
            lineMerge.N = 1        /* Will merge this line */
            PARSE VAR ALINE lSeg.N =(P) JUNK +7 Case +1 rSeg.N


            SELECT                 /* -----Start Select ----- */
               WHEN Case = '1' THEN
```

```
                  DO
                     Name_case.N = 1     /* uppercase   */
                  END
                WHEN Case = '2' THEN
                  DO
                     Name_case.N = 2     /* lowercase  */
                  END
                WHEN Case = '3' THEN
                  DO
                     Name_case.N = 3     /* first letter uppercase */
                  END
                OTHERWISE
                  DO
                     Name_case.N = 1     /* case defult to upper  */
                  END
             END                    /* ----- End of select ----- */

           END     /* ----- End of DO  for If (p \= 0) then ------*/
        ELSE       /* ---- ELSE of         If (p \= 0) then ------*/
           DO
             lSeg.N = ALINE         /* save the line unchanged   */
             lineMerge.N = 0        /* set lineMerger flag to 0  */
           END

      END          /*- End of DO for IF (RETURN_CODE1 \= 0) THEN -*/

    ELSE
      DO
         IF (RC \= 2) THEN
         DO
         SAY "TEMPLATE FILE "||(INFILE1)||" READ ERROR !"
         SAY 'RC=' RC
         EXIT 3
         END
         ELSE NOP
      END
 END
     nTotal = N                   /* TOTAL TEMPLATE LINES */
     N = 0               /* RACF ldif Input file counter */
     R = 0               /* # of racfid= lines found     */
     rPart  = ''         /* init */

 say ''
 say " Start to read racfid list file ...  "
 say ''

 /*----------------------------------------------------------------*/
 /*  step3:  read in the RACF racfid list file                     */
 /*----------------------------------------------------------------*/
 RETURN_CODE1 = 0          /* INITIALLZE RETURN CODE               */

DO WHILE (RETURN_CODE1 \= EOFFLAG1) /* loop while not end-of-file  */

   'EXECIO 1 DISKR INDD2'          /* read 1 line from table file */
   RETURN_CODE1 = RC               /* save execio rc              */

   IF RETURN_CODE1 = 0 THEN        /* get a line ok?              */
      DO                           /* yes                         */
        N = N + 1                  /* Total line counter          */
        PARSE PULL ALINE
        P1 = POS(TOKEN1,ALINE,1)   /* FIND first 'racfid='  */
        IF (P1 \=0 ) THEN          /*  Yes                  */
          DO
            PARSE VAR ALINE JUNK =(P1) rPart
            P2 = POS(',',rPart,1)
            PARSE VAR rPart JUNK2 +7 name =(P2) JUNK1
            RACFID = STRIP(name)   /* clean the blanks */
            uname = RACFID         /* save racfid to uname  */
             R = R + 1           /* 'racfid=' line counter */

/* --> WRITE OUT THE MERGED Output to file */

        DO K = 1 TO nTotal       /* loop though all template lines */
          ALINE1 = ''            /* Init the output line  */
          IF (lineMerge.K \= 0) THEN
            DO
              Case = Name_case.K     /* User Name case type */
              SELECT             /*  Strat of the select   */
                WHEN Case = 1 THEN
                  DO
                    Upper uname
                  END
                WHEN Case = 2 THEN
                  DO
                     s3 = uname      /* change it to lower case */
                     uname = TRANSLATE(s3,strLower,strUpper)
                  END
                WHEN Case = 3 THEN
                  DO
                     PARSE VAR uname s1 +1 s2
                     Upper s1
                                       /* change it to lowercase  */
                     s3 = TRANSLATE(s2,strLower,strUpper)
                     uname  = (s1)||(s3)
                  END
                OTHERWISE
                  DO
```

```
                     Upper uname
                  END

              END      /* ----- End of select ----- */

              ALINE1 = lSeg.K||uname||rSeg.K
          END          /* ------ End of DO ---- */
        ELSE
          DO
              ALINE1 = lSeg.K
          END          /* ------ End of ELSE ---- */

        PUSH ALINE1
        'EXECIO 1 DISKW OUTDD1'  /* write aline to tsscmd file */

        END      /* ----- End of DO K = 1 ----- */

        ALINE1 = ''              /* Write a blank line */
        PUSH ALINE1
        'EXECIO 1 DISKW OUTDD1'

      END       /* ----- End of IF (P1 \= 0) THEN DO ----*/
      ELSE NOP
     END        /* ===== End of IF (RETTURN_CODE1 = 0) =====*/
   ELSE
     DO
        IF (RC \= 2) THEN
        DO
          SAY "RACF FILE "||(INFILE2)||" LINE# "||(N)||" ERROR !"
          EXIT 4
        END
        ELSE NOP
     END
 END            /* ------- End of do while ------- */

 nInlines = N
 nRacfids = R

say ''
say "Total Input File lines: "||(nInlines)||" "
say "Total racfid= lines   : "||(nRacfids)||" "
say ''

"EXECIO 0 DISKR INDD1 (FINIS"           /* close the input file   */
"EXECIO 0 DISKR INDD2 (FINIS"           /* close the input file   */
"EXECIO 0 DISKW OUTDD1 (FINIS"          /* close the output file   */
"FREE FI(INDD1)"
"FREE FI(INDD2)"
"FREE FI(OUTDD1)"
```

```
exit
```

D

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

8.2.2 Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

`ftp://www.redbooks.ibm.com/redbooks/SG246992`

Alternatively, you can go to the IBM Redbooks Web site at:

`ibm.com/redbooks`

Select the Additional materials and open the directory that corresponds with the redbook form number, SG246992.

8.2.3 Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	Description
sg246992.zip	The zip file convldif.zip contains a REXX program convldif.rexx, a template file template.ldif, a exported

sample RACF users list file, racfid.input and a sample merged output file merged.output.

8.2.4 System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	10MB minimum
Operating System:	Windows/UNIX or AIX®
Processor:	500 Mhz or higher
Memory:	512MB

8.2.5 How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. Those files are used as an example for converting an exported RACF userid list file into an ldif format file for the Portal LDAP Server.

Refer to 7.5.6, “Mass-exporting RACF users to LDAP” on page 313 for instructions on setup and the usage of the sample REXX program.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 343. Note that some of the documents referenced here may be available in softcopy only.

- *e-business Cookbook for z/OS: Technology Introduction,* SG24-5664
- *A Portal Composite Pattern Using WebSphere V4.1,* SG24-6869
- *WebSphere Studio Application Developer Programing Guide,* SG24-6585
- *IBM WebSphere Portal V4 Developer’s Handbook,* SG24-6897
- *From code to deployment: Connecting to CICS from WebSphere for z/OS,* REDP0206
- *Enabling High Availability e-Business on IBM eServer zSeries*, SG24-6850
- *LDAP Server Administration and Usage Manual*, SC24-5923

Other publications

These publications are also relevant as further information sources:

- *z/OS Communications Server IP Configuration Guide*, SC31-8775-01
- *z/OS Communications Server: SNA Network Implementation*, SC31-8777
- *DB2 Data Sharing: Planning and Administration*, SC26-9935
- *WebSphere Application Server Version 4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834-06
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*, SA22-7835

Online resources

These Web sites and URLs are also relevant as further information sources:

- SUN J2EE information site

  `http://java.sun.com/j2ee/overview.html`
- WebSphere Developer Domain

  `http://www7b.software.ibm.com/wsdd/`
- WPCP help pages

  `http://publib.boulder.ibm.com/wcmid/mp/v42/helpsystem/en/index.html`
- Installing Prerequisite Software for the IBM Portal Toolkit Plug-in for WebSphere Studio Application Developer

  `http://www7b.software.ibm.com/wsdd/library/techarticles/0210_phillips/phillips2.html`
- Developing and Debugging Portlets Using the IBM Portal Toolkit Plug-in for WebSphere Studio Application Developer

  `http://www7b.software.ibm.com/wsdd/library/techarticles/0210_phillips/phillips.html`
- WPCP Getting Started Guide

  `http://publib.boulder.ibm.com/wcmid/mp/v42/helpsystem/en/gspdf.pdf`
- WebSphere Portal Product Documentation

  `http://www7b.software.ibm.com/wsdd/zones/portal/proddoc.html`
- README file for PTF2

  `http://publib.boulder.ibm.com/pvc/wp/415/zos/en/readme41ptf2_zos.html`
- README for WebSphere Portal installation

  `http://publib.boulder.ibm.com/pvc/wp/415/zos/en/readme41ptf2_install_zos.html`
- z/OS UNIX tools page

  `http://www-1.ibm.com/servers/eserver/zseries/zos/unix/bpxa1ty2.html`
- Portlet API specification

  `http://jcp.org/en/jsr/detail?id=168`
  `http://jcp.org/jsr/detail/168.jsp`
- Portlet API description

  `http://publib.boulder.ibm.com/pvc/wp/current/ena/en/InfoCenter/index.html`
- Portlet Development Guide

http://www7b.software.ibm.com/wsdd/zones/portal/portlet/portletdevelopmentguide.html

- Viewing the WebSphere Portal V1.4 Portlets page

  http://www7b.software.ibm.com/wsdd/library/techarticles/0301_son/0301_son.html
- WebSphere Studio Application Developer homepage

  http://www-3.ibm.com/software/awdtools/studioappdev/
- Developing and Invoking a Servlet from a Portlet using WebSphere Studio

  http://www7b.software.ibm.com/wsdd/techjournal/0302_konduru/konduru.html
- WebSphere Portal catalog

  http://www-3.ibm.com/services/cwi/portal/_pagr/105/
- CICS Trader application

  http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0206.html?Open
- Additional materials for this redbook can be found at:

  ftp://www.redbooks.ibm.com/redbooks/SG246992

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Index

IBM

Redbooks

WebSphere Portal on z/OS

(0.5" spine)
0.475"<->0.875"
250 <-> 459 pages

WebSphere Portal on z/OS

Portal architecture and installation

Portlet deployment

Installation tips and problem determination

This IBM Redbook will help you install, tailor, and configure the PTF2 level of WebSphere Portal Server for z/OS product. We discuss architectural, installation, configuration, administration, security and problem determination issues. We show you how to deploy portlets and how to convert a "normal" WebSphere application into a portlet.

Some of the key features of the IBM WebSphere Portal offering are that it delivers a single, universal point of access that is integrated, highly customizable and scalable to interact with key applications, content, people, and business processes. Also, it offers numerous portlets for e-mail, calendars, syndicated news, industry applications and many other functions.

The WebSphere Portal for Multiplatforms includes three editions: Portal Enable, Portal Extend, and Portal Experience.

This redbook covers the WebSphere Portal for z/OS and OS/390 product, which is based on the WebSphere Portal V4.1 Enable edition.

SG24-6992-00 ISBN 0738499382